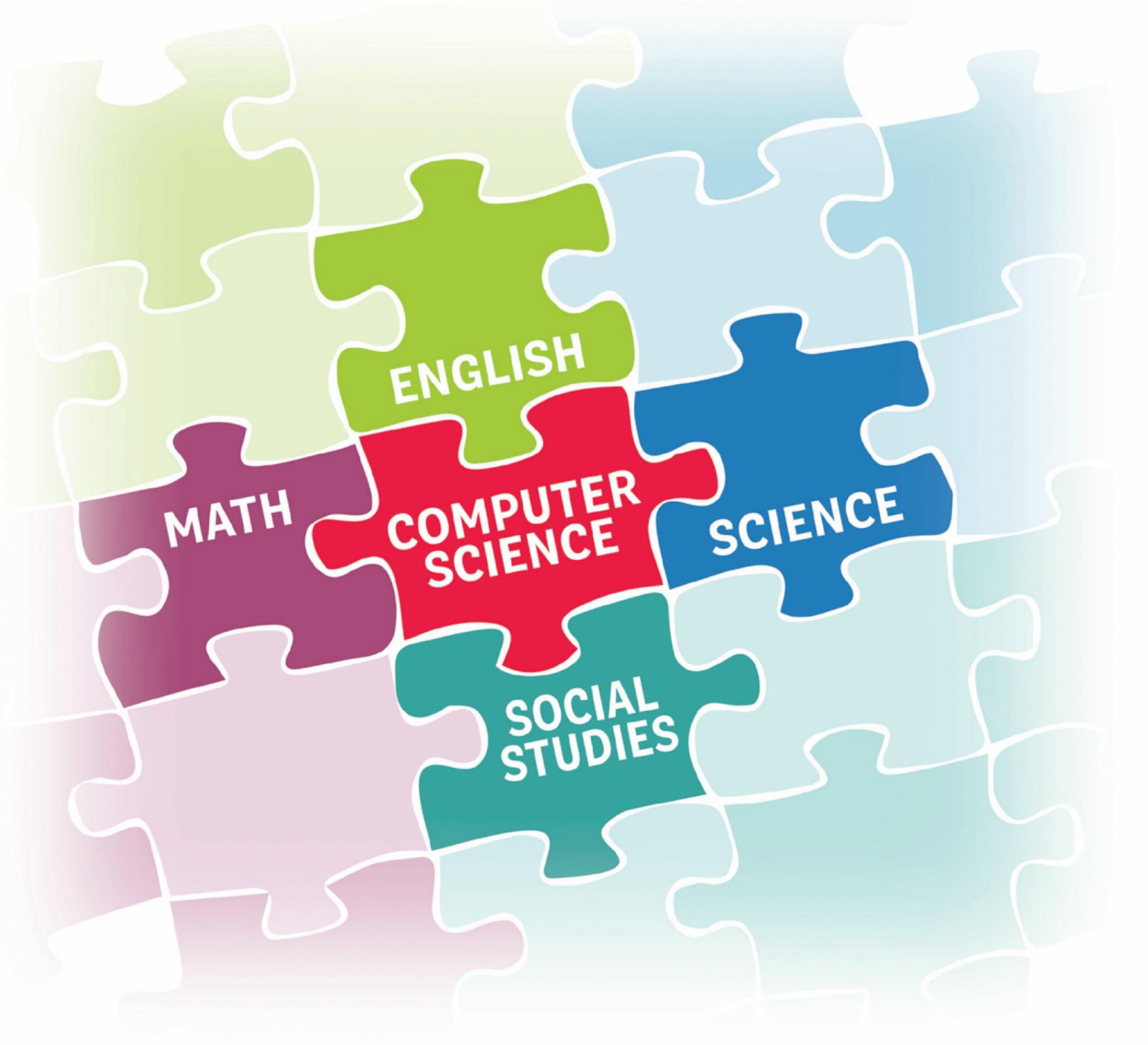


2022

Elementary Integration Guide

FOURTH GRADE

MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

Acknowledgements

Elementary Task Force

Melissa Atkins, Instruction Technologist, Lamar County School District
Kacy Baggett, Teacher, Rankin County School District
Heather Barry-Fenster, Teacher, Lowndes County School District
Brittany Boatman, Teacher, Houston School District
Kimberly Brammer, Teacher, Pascagoula-Gautier School District
Michelle Carter, Teacher, Picayune School District
Jana Chao, Teacher, Clinton Public School District
Lerenda Dixon, Teacher, McComb School District
Samantha Elizondo, Teacher, Tupelo Public School District
Angie Frazier, Teacher, Rankin County School District
Tammy Hale, Teacher, Tate County School District
Kayla Hathcock, Teacher, Amory School District
Ashley Hawkins, Teacher, Biloxi Public School District
Vicky Johnson, Teacher, Franklin County School District
Dinah Lachney, Teacher, Madison County School District
Ashley Matthews, Administrator, Lowndes County School District
Kayla Moore, Teacher, Enterprise School District
Olivia Moore, Teacher, DeSoto County School District
Beth Neese, Teacher, Western Line School District
Hannah Padgett, Teacher, Tate County School District
Dr. Lee Pambianchi, Administrator, Rankin County School District
Kristen Phillips, Teacher, Oxford School District
Brittney Price, Teacher, South Panola School District
Melissa Sundberg, Teacher, Ocean Springs School District
Melissa Tingle, Teacher, Lauderdale School District
Susie Williams, Curriculum Coordinator, Leland School District

Mississippi Department of Education Team

Wendy Clemons, executive director, Office of Secondary Education
Louella Webster, supervisor for computer science/STEM

Center for Cyber Education Team

Shelly Hollis, Director
Lizzie Brandon, Project Manager
Amanda Taylor, Project Manager

Research and Curriculum Unit Team

Brock Turnipseed, Marketing and Communications Manager
Chris McMillen, Communications Coordinator
Heather Craig, Editor
Will Graves, Project Coordinator

Funding for the development of this guide was provided by:

c spire

Introduction

In March 2021, The Mississippi Computer Science and Cyber Education Equality Act (House Bill 633) was passed requiring all districts to offer computer science content and courses by the 2024-2025 school year. The bill allows for a phased-in approach as listed below:

2022-2023: All middle schools offer at least one (1) course in computer science, and 50% of elementary schools offer a minimum of one (1) hour of instruction in computer science each week at each grade level.

2023-2024: All elementary schools offer a minimum of one (1) hour of instruction in computer science each week at each grade level, and 50% of high schools offer at least one (1) course in computer science.

2024-2025: All schools will offer instruction in computer science.

To make the integration of computer science content as seamless as possible for elementary teachers, a task force of elementary teachers, principals, the Mississippi Department of Education, and the Mississippi State University Center for Cyber Education was formed to write an integration guide for each grade level, kindergarten through fifth grade. These guides provide plans for a minimum of 40, 60-minute lessons covering six computer science topics: coding, robotics, digital literacy, digital citizenship, keyboarding, and unplugged activities.

Each guide contains a breakdown of content by integrated subjects, content by computer science topics, and a calendar/pacing guide. Teachers may choose to start at the beginning and teach each lesson once a week in chronological order or teach the lesson that integrates with another core subject topic at a more relevant time. In addition to a lesson overview and links to required resources, each lesson plan maps to a Mississippi Computer Science Standard and a core subject area standard. A suggestion on how to break the lesson into smaller segments to be covered throughout the week is also provided in the "Time needed" section.

There are several resources available in each integration guide. Some may require the creation of accounts, but all resources referenced are free. The pacing guide notes lessons requiring account creation so teachers can plan ahead. A list of sites used is provided for technology departments to whitelist or unblock. All resources may be used on any internet-capable device, including Chromebooks and tablets.

Resources	
Computing resources	- Code.org CS Fundamentals - 4th Grade: Course E - Common Sense Digital Media - Scratch - Kahoot - CS Unplugged
CS4MS website materials	- 2018 Mississippi Computer Science Standards - CS4MS Website
Keyboard practice	- Astro Bubbles Keyboard Practice - Dance Mat Typing Site - Nitrotype - Typing.com
Teacher/student accounts	- Code.org - Common Sense Digital Media - Scratch - Kahoot - CS Unplugged
For help with this guide	- Contact Mississippi State University's Center for Cyber Education: www.tinyurl.com/ccehelpdesk

Contents by Integrated Subjects

English Language Arts

Math

Science

Social Studies

Contents by Topics

Coding

Digital Citizenship

Digital Literacy

Keyboarding

- Use as a filler to build typing fluency
  - Resources: NitroType, Typing Club (a free product of EdClub)

Robotics

Unplugged

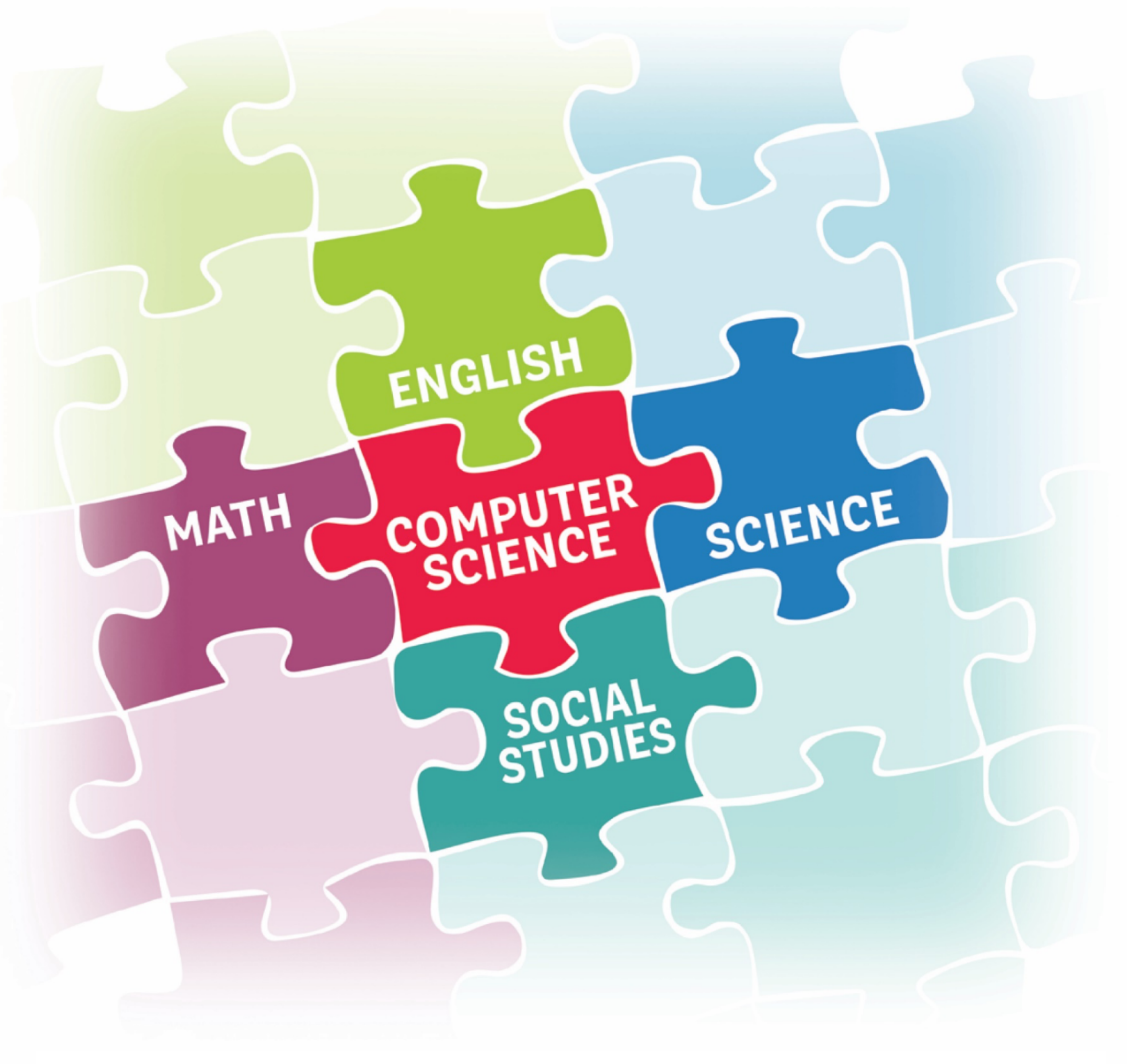
Calendar/Pacing Per Week:

⟶Teachers will need to create a FREE **teacher and/or student account** (See the notes section of the lesson.).

Week	Title	Topics	CS Standard	Standard	Subject Integrated
1	Keyboarding Basics	Digital Citizenship	CS.1B.1	None	None
2	Code.org: Graph Paper Programming → Account creation needed	Unplugged	AP.1B.4	RI.4.2 RL.4.2	ELA
3	Code.org: Introduction to Online Puzzles → Account creation needed	Unplugged	AP.1B.4	RI.4.3	ELA
4	Code.org: Be A Super Digital Citizen	Unplugged	NB.1B.2	CI.4.3	ELA
5	Programming Using Maps	Unplugged	AP.1B.1a AP.1B.5 AP.1B.4a	G.4.3.2	Social Studies
6	Code.org: Private and Personal Information	Digital Citizenship	NB.1B.2 NI.1B.2a	CI.4.3	Social Studies: MS Studies
7	Code.org: Swimming Fish With Sprite Lab	Coding	AP.1B.5	W.4.3a	Writing
8	Code.org: Hello World	Coding	AP.1B.5	RI.4.5	ELA
9	Code.org: Mini Project: Career Journeys	Coding	AP.1B.14 IC.1B.21 NI.1B.05	W.4.7 W.4.9 SL.4.4	ELA
10	Code.org: Game Design	Coding	AP.1A.11 AP.1B.10	L.4.1d	ELA
11	Code.org: Variables as Score	Coding	AP.1A.11 AP.1B.09 AP.1B.10	L.4.1.f	ELA
12	Code.org: Environment and Players	Coding	AP.1A.11 AP.1B.09 AP.1B,10	G.4.3.2	Social Studies
13	Code.org: Game Jam Day 1: Create	Coding	IC.1B.4 NI.1B.2	P.4.6A.6	Science
14	Code.org: Game Jam Day 2: Play	Coding	AP.1B.13 DA.1B.07 DA.1B.06	L.4b	ELA
15	Interactive Presentation → Account creation needed	Coding	AP.1B.4 AP.1B.6	P.4.6a P.4.6b P.4.6.c	Science

16	Code.org: Songwriting	Unplugged	AP.1B.1 AP.1B.1a AP.1B.4 AP.1B.4a	H.4.1	Social Studies
17	Code.org: Sequential Order Using Scratch	Coding	AP.1B.3	RL.4.1 RI.4.1	ELA
18	Code.org: Functions in Music Lab	Coding	AP.1B.08 AP.1B.10 AP.1B.11 AP.1B.15	H.4.4.1	Social Studies
19	Research a Scientist Project	Digital Literacy	AP.1B.2 AP.1B.3	P.4.6c.3 W.4.7	Science, Writing
20	Civil Rights History	Digital Literacy	AP.1B.3a	4.MS.8	Social Studies
21	Animating the Steps of Long-Division	Coding	AP.1B.1 AP.1B.4	4.NBT.6	Math
22	Theme, Main Idea, and Details	Coding	AP.1B.10a 1B.AP.10	RL.4.2 RI.4.2	ELA
23	Code.org: Functions in Minecraft	Coding	AP.1B.08 AP.1B.10 AP.1B.12 AP.1B.15	4.MD.5	Math
24	Life Cycles	Digital Literacy	AP.1B.10a AP.1B.10	L.4.2	Science
25	CSFirst: Code Your Hero	Coding	AP.1B.3a AP.1B.6b	4.W.1	Writing
26	Subtracting Across Zeros	Digital Literacy	AP.1B.1 AP.1B.4	4.NBT.1 4.NBT.4	Math
27	Literal and Figurative Language	Coding	AP.1B.10 AP.1B.11	RL.4.4	ELA
28	Code.org: Functions With Artist	Coding	AP.1B.08 AP.1B.10 AP.1B.11 AP.1B.12 AP.1B.17	CI.4.3.1	Social Studies
29	Code.org: Conditionals in Minecraft Voyage Aquatic	Coding	AP.1B.3	4.NBT.5	Math
30	Weather Safety	Coding	AP.1B.10 AP.1B.11	E.4.9C.5	Science
31	Famous Mississippians	Coding	AP.1B.3a AP.1B.6b	CR.4.1.1	Social Studies
32	Points, Lines, Segments, Rays, and Angles	Robotics	AP.1B.3	4.G.1	Math

33	Digital Sharing Animation	Digital Citizenship	IC.1B.21 NI.1B.5	RI.4.9	ELA
34	Maze Mat Race	Robotics	AP.1B.1 AP.1B.4	RL.4.1 RL.4.2 RL.4.3 RL.4.4	ELA
35	Energy Sources	Coding	AP.1B.5 AP.1B.6	E.4.10.1 E.4.10.2	Science
36	Code.org, Conditionals With the Farmer	Coding	AP.1B.10 AP.1B.11 AP.1B.17	G.4.1.1 G.4.2.3	Social Studies
37	Angle Measures	Robotics	AP.1B.4	4.MD.7	Math
38	Text Structures: Coding	Coding	AP.1B.4	RI.4.5	ELA
39	Text Structures: Robotics	Robotics	AP.1B.1 AP.1B.4	RI.4.5	ELA
40	Code.org, Designing for Accessibility	Coding	AP.1B.12 AP.1B.13 CS.1B.3 IC.1B.18 IC.1B.19 IC.1B.20	RL4.1	ELA
41	Code.org, Digital Sharing	Unplugged, Digital Citizenship	AP.1B.14 IC.1B.21	W.4.6	Writing



Lessons and Activities

FOURTH GRADE

Week 1: Keyboarding Basics Lesson

Lesson overview:	**Purpose:** In this keyboarding lesson, students will practice exhibiting the proper techniques for keyboarding using various instructional engagement media, such as videos and an online keyboarding game with various levels of keyboarding-skill mastery. **Lesson:** ● Introduction ○ Follow the prompts on the Intro to Keyboarding – Teacher-Created Lesson Plan _ Common Sense Education.pdf ○ Proper Sitting Postures on Computer - video ● Keyboarding Practice ○ Keyboard Finger Guide ○ Dance Mat Typing Site ● Digital Citizenship ○ Keeping Games Fun and Friendly
Lesson links/resources:	Lesson plan ● Intro to Keyboarding – Teacher-Created Lesson Plan _ Common Sense Education.pdf ● Proper Sitting Postures on Computer - video ● Dance Mat Typing Site ● Keyboard Finger Guide Digital citizenship lesson: ● Keeping Games Fun and Friendly
CS standards addressed:	Students will be able to: ● Identify home row keys ● Demonstrate proper sitting posture for typing ● Identify ways to develop positive interactions while gaming online Standards: ● **CS.1B.1—**Describe how internal and external parts of computing devices function to form a system.
Time needed:	**Total Time: 60 min** ● Introduction Video and Discussion **5 min** ● Keyboarding Practice **30 min** ● Digital Citizenship Activity **15 min** ● Wrap-up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	None
Other standards addressed:	None
Vocabulary:	Home row keys: The home keys are where you place your fingers when you are learning to type. The home keys include F, D, S, and A on the left

	of the keyboard, and J, K, L, and ; (semicolon) on the right of the keyboard.
Notes:	

Week 2: Code.org, Course D, Lesson 2—Graph Paper Programming

Lesson overview:

Purpose:
The goal of this activity is to build critical thinking skills and excitement for the course, while introducing some of the fundamental programming concepts that will be used throughout the course. By introducing basic concepts like sequencing and algorithms to the class in an unplugged activity, students who are intimidated by computers can still build a foundation of understanding on these topics. In this lesson, students will learn how to develop an algorithm and encode it into a program.

Lesson:

- Introduction
  - Play one of the three videos to show students the type of things robots can do.
    - **Asimo by Honda** (3:58)
    - **Egg Drawing Robot** (3:15)
    - **Dancing Lego Robot** (1:35)
- Graph Paper Programming
  - In this activity, students will act as both programmers and robots, coloring in squares according to programs that they have written for one another.
- Reflection
  - What was today's lesson about?
  - How did you feel during today's lesson?
  - Draw another image that you could code. Can you write the program that goes with this drawing?
  - What other types of robots could we program if we changed what the arrows meant?
- Assessment
  - Independent practice writing and reading a program
- Keyboarding
  - If time remains, have students use resources from above to practice keyboarding.

Lesson links/resources:

- Course D Lesson 2: Graph Paper Programming
- Common Sense: Our Responsibilities Online
- **Asimo by Honda** (3:58)
- **Egg Drawing Robot** (3:15)
- **Dancing Lego Robot** (1:35)

CS standards addressed:

Students will be able to:

- Explain constraints of translating problems from human language to machine language
- Reframe a sequence of steps as an encoded program

Standards:

- **AP.1B.4**—Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

Time needed:

Total Time: 60 min

- Warm Up/Introduction **10 min**
- Main Activity **20 min**
- Wrap Up **15 min**

	- Digital Citizenship Topic **15 min**
Materials needed:	Teachers: - Smartboard/projector with sound - Graph Paper Programming - Lesson in action video - Graph Paper Programming - Worksheet answer key - Graph Paper Programming - Assessment answer key Students: - Student devices with access to the internet - Graph Paper Programming - Activity worksheet - Graph Paper Programming - Unplugged video (Download) - Graph Paper Programming – Assessment
Subject integrated:	ELA
Other standards addressed:	- **RI 4.2**—Determine the main idea of a text and explain how it is supported by key details; summarize the text. - **RL4.2**—Determine a theme of a story, drama, or poem from details in the text; summarize the text.
Vocabulary:	Algorithm: A list of steps to finish a task Program: An algorithm that has been coded into something that can be run by a machine
Notes:	→ Teachers will need to create FREE teacher and/or student accounts https://studio.code.org/users/sign_in

Week 3: Code.org, Course D, Lesson 3—Introduction to Online Puzzles

Lesson overview:	**Purpose:** Every classroom has a spectrum of understanding for every subject. Some students in your class may be computer wizards, while others have not had much experience at all. In order to create an equal playing (and learning) field, Code.org has developed this "Ramp Up Stage" for Course D. This can be used as either an introduction or a review of how to use Code.org and basic computer science concepts. This stage covers all prerequisites needed to start Course D. **Lesson:** ● Introduction ○ Vocabulary words ● Preview of online puzzles as a class ○ Bring the unplugged activity from the previous lesson to an online platform. ● Online Puzzles ○ Introduction to programming with blocks ● Reflection ● Digital Citizenship (Optional) ○ The words we choose
Lesson links/resources:	● Course D Lesson 3: Introduction to Online Puzzles ● Common Sense: The Words We Choose
CS standards addressed:	Students will be able to: ● Break down a long sequence of instructions into the largest repeatable sequence ● Modify an existing program to solve errors ● Order movement commands as sequential steps in a program Standards: ● **AP.1B.4**—Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **10 min** ● Bridging Activities **10 min** ● Main Activity **30 min** ● Wrap Up **10 min** ● Digital Citizenship Topic (optional) **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet ● Pair Programming - student video
Subject integrated:	ELA
Other standards addressed:	**RI4.3**—Explain events, procedures, ideas, or concepts in a historical, scientific, or technical text, including what happened and why, based on specific information in the text.

Vocabulary:	Bug: Part of a program that does not work correctly Debugging: Finding and fixing problems in an algorithm or program Loop: The action of doing something over and over again Program: An algorithm that has been coded into something that can be run by a machine Programming: The art of creating a program Technical text: Instructions telling us how to use or do something
Notes:	→Teachers will need to create free teacher and/or student accounts (when applicable) at https://www.commonsense.org/education/

Week 4: Code.org, Course E, Lesson 1—Be A Super Digital Citizen

Lesson overview:	**Purpose:** Common Sense's Digital Citizenship Curriculum addresses six core topics, based on the latest research on children, media, and technology. This lesson focuses on Cyberbullying, Digital Drama & Hate Speech. Students take on these tough topics and play the active role of upstanders to build positive, supportive online communities and combat online cruelty. **Lesson:** ● Secret Superhero ○ Key vocabulary ● Being an Upstander ○ Discuss the meaning of being a cyberbully ○ What Would a Super Digital Citizen Do? Student Handout ● Create Your Digital Superhero ○ Digital Citizen Superhero Student Handout ○ Marvel's Create Your Own Superhero ● Save the Day Comic Strip ○ Classroom-Friendly Websites and Apps for Making Comics ● Keyboarding Practice If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	Course E Lesson 1: Be a Super Digital Citizen
CS standards addressed:	Students will be able to: ● Define and recognize various forms of cyberbullying ● Explain ways that cyberbullying can be addressed Standards **NB.1B.2** Discuss real-world cybersecurity problems and how personal information can be protected (cybersecurity).
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **5 min** ● Main Activity: Learn **10 min** ● Main Activity: Create **15 min** ● Wrap Up **15 min** Keyboarding (See "other" resources) **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound ● Be A Super Digital Citizen: Lesson Slides Students: ● Student devices with access to the internet Be A Super Digital Citizen: Super Digital Citizen (Download)
Subject integrated:	Social Studies
Other standards addressed:	**CI:4.3**—Identify rights and responsibilities as a citizen of your community or state.

Vocabulary:	Cyberbullying: Using digital devices, sites, and apps to intimidate, harm, and upset someone Digital citizen: Someone who uses technology responsibly to learn, create, and participate Upstander: A person who supports and stands up for someone else
Notes:	

Week 5: Programming Using Maps

Lesson overview: Unplugged	**Purpose:** In this context-setting lesson, students will play a game intended to get them thinking about sequential instructions. **Lesson:** ● The teacher will provide the students with a map of Mississippi on chart paper or a large poster board. The students will locate their city and then give directions to points of interest throughout the state (Jackson, Ross Barnett Reservoir, MS Gulf Coast, Natchez, Vicksburg, etc.) or to the hometowns of famous Mississippians. ● The students will use the square sticky notes as a "grid" to mark how many moves should be made in order to reach their location from their starting point. ● The students may use the coding cards (see resources) to map out their path. ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	Coding Cards (arrows)
CS standards addressed:	Students will be able to: ● Begin to make connections about a new way to write programs Standards: ● **AP.1B1a** students should be able to look at different ways to solve the same task and decide which would be the best solution. ● **AP.1B.5** Modify, remix, or incorporate portions of an existing program into one's own work to develop something new or add more advanced features. ● **AP.1B.5a** Students should be able to modify and/or reuse portions of an existing program into their own work to create something new.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **10 min** ● Main Activity **20 min** ● Wrap Up **15 min** ● Keyboarding Practice **15 min**
Materials needed:	Teachers: ● Chart paper ● Markers ● Map of Mississippi ● Small square sticky notes Students: ● Chart paper ● Markers ● Map of Mississippi ● Small square sticky notes
Subject integrated:	Social Studies

Other standards addressed	**G.4.3.2**—Recognize maps, graphs, and other representations of Mississippi.
Vocabulary:	Behavior: An action that a sprite performs continuously until it is told to stop Algorithm: A list of steps to finish a task
Notes:	

Week 6: Code.org, Course E, Lesson 2—Private and Personal Information

Lesson overview:	**Purpose:** Common Sense's Digital Citizenship Curriculum addresses six core topics based on the latest research on children, media, and technology. This lesson focuses on privacy and security. Students learn how to protect personal information and gain a deeper understanding of their data privacy rights so they can advocate for themselves and others. **Lesson:** ● Stand Up, Sit Down ○ Key Vocabulary ● Why Do People Share? ○ Learn how much is safe to share online ● Private or Personal? ○ Learn what is safe to share online ● Assessment ○ Private and Personal Information: Lesson Quiz ● Exit Ticket ○ Private and Personal Information: Exit Ticket
Lesson links/resources:	Course E Lesson 2: Private and Personal Information
CS standards addressed:	Students will be able to: ● Explain the difference between private and personal information ● Explain why it is risky to share private information online ● Identify the reasons why people share information about themselves online Standard: ● **NI.1B.2**—Discuss real-world cybersecurity problems and how personal information can be protected. ● **NI.1B.2a**—Students should be able to explain what passwords are, why we use them, and use strong passwords to protect devices and information from unauthorized access.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **10 min** ● Analyze: Why people share? **10 min** ● Analyze: Private or personal? **15 min** ● Assessment **15 min** ● Wrap Up/Exit Ticket **10 min**
Materials needed:	For the teachers ● Smartboard/projector with sound ● Private and Personal Information: Lesson Quiz-Answer key ● Private and Personal Information: Lesson Slides-Slide deck For the students ● Student devices with access to the internet ● Private and Personal Information: Exit Ticket-Student handout ● Private and Personal Information: Lesson Quiz-Form

	- Private and Personal Information: Private and Personal Information-Student video (Download)
Subject integrated:	Social Studies
Other standards addressed:	**CI:4.3**—Identify rights and responsibilities as a citizen of your community or state.
Vocabulary:	Hardwired: Something you are born with Personal Information: Information about you that cannot be used to identify you because it is also true for many other people (e.g., your hair color or the city you live in) Private Information: Information about you that can be used to identify you because it is unique to you (e.g., your full name or your address) Register (Online): To enter your information to sign up and get access to a website or app
Notes:	

Week 7: Code.org, Course E, Lesson 3—Swimming Fish With Sprite Lab

Lesson overview:	**Purpose:** This lesson is designed to introduce students to the core vocabulary of Sprite Lab and allow them to apply concepts they learned in other environments to this tool. By creating a fish tank, students will begin to form an understanding of the programming model of this tool and explore ways they can use it to express themselves. **Lesson:** ● Introduction ○ What is a sprite? ○ A sprite is controlled by a program ● Swimming Fish Teacher Sandbox ○ Model writing programs for the sprite ● Swimming Fish with Sprite Lab ○ Students will move through the lessons to create their own fish tank ● Reflection ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	Course E Lesson : Swimming Fish With Sprite Lab
CS standards addressed:	Students will be able to: ● Create new sprites and assign them costumes and behaviors ● Define "sprite" as a character or object on the screen that can be moved and changed Standards: ● **AP.1B.5**—Modify, remix, or incorporate portions of an existing program into one's own work to develop something new or add more advanced features.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **10 min** ● Bridging Activity **10 min** ● Main Activity **20 min** ● Wrap Up **15 min** ● Keyboarding Practice **5 min**
Materials needed:	Teachers: ● Smartboard/projector with sound ● Swimming Fish Teacher Sandbox - Programming level Students: ● Student devices with access to the internet ● Sprite Lab Documentation – Resource
Subject integrated:	Writing
Other standards addressed:	**W.4.3**—Write narratives to develop real or imagined experiences or events using effective technique, descriptive details, and clear event

	sequences.
Vocabulary:	Behavior: An action that a sprite performs continuously until it is told to stop Sprite: A graphic on the screen with a location, size, and appearance
Notes:	

Week 8: Code.org, Course E, Lesson 4 - Hello World

Lesson overview:	**Purpose:** This lesson offers a great introduction to events in programming and even gives students a chance to show creativity! At the end of the lesson, students will be presented with the opportunity to share their projects. **Lesson:** ● Introduction ○ Setting the Stage ○ A series of events ○ New vocabulary ● Hello World ○ Video: Hello World: Welcome to Sprite Lab ○ Skill building ○ Video: Hello World: Programming with Events ○ Skill Building ○ Video: Hello World: Do What You Want ○ Free play ● Reflection ● Free Play ○ If time remains, students can free play in their Sprite Lab to experiment different programs
Lesson links/resources:	Course E Lesson 4: Hello World
CS standards addressed:	Students will be able to: ● Create an interactive animation using sprites, behaviors, and events ● Identify actions that correlate to input events Standards: ● **AP.1B.5—**Modify, remix, or incorporate portions of an existing program into one's own work to develop something new or add more advanced features.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **5 min** ● Main Activity **30 min** ● Wrap Up **10 min** ● Free Play **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet ● Sprite Lab Documentation – Resource
Subject integrated:	ELA
Other standards addressed:	**RI 4.5**—Describe the overall structure (e.g., chronology, comparison, cause/effect, problem/solution) of events, ideas, concepts, or information in a text or part of a text.
Vocabulary:	Event: An action that causes something to happen

Week 9: Code.org, Course E, Lesson 5 – Mini-Project: Career Journeys	
Lesson overview:	Purpose: This lesson introduces students to the wide range of careers that involve computer science and artificial intelligence. Through interactive videos and guided exploration, students learn about real-world jobs that use CS. They are then guided step by step to build their own coding projects using Sprite Lab to present an interactive scene that relates to a career or subject they may be interested in. The lesson wraps up with students reflecting on what they've learned and how it connects to their future goals. **Lesson:** ● Introduction - Show CS is Everything Video. **Say**: Think about careers that work on ideas and things in the video. What did you like about this video? What are some examples of careers that use CS? ● Career Exploration - Set the timer for four minutes so students know when to move to the next level. Code Projects – Use the slides to discuss tips for using the Sprite Lab and introduce sprite and event as the two key CS concepts they will use in their projects. ● Research – Students can research careers and write a summary of the career they think they would enjoy persuing. ● Share Projects ● Reflection
Lesson links/resources:	Course E, Lesson 5 – Mini-Project: Career Journeys
CS standards addressed:	Students will be able to: • Design and build an interactive project • Explore and describe careers that use computer science • Practice basic coding skills • Present a project and explain the coding skills used Standards: • **1B-AP-14** - Observe intellectual property rights and give appropriate attribution when creating or remixing programs. • **1B-IC-21** - Use public domain or creative commons media and refrain from copying or using material created by others without permission.

	- **1B-NI-05** - Discuss real-world cybersecurity problems and how personal information can be protected.
Time needed:	- Total Time: 45 min - Warm Up **- 5 min** - Career Exploration – **20 min** - Code your Project **– 15 min** - Wrap Up **- 5 min**
Materials needed:	Teachers: - Smartboard/projector with sound - Slides: **Helpful Hints in Sprite Lab** - Resource (**Download**) Students: - Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	W.4.7 Conduct short research projects that build knowledge through investigation of different aspects of a topic. W.4.9 Draw evidence from literary or informational texts to support analysis, reflection, and research. a. Apply Grade 4 Reading Standards to literature (e.g., “Describe in depth a character, setting, or event in a story or drama, drawing on specific details in the text [e.g., a character’s thoughts, words, or actions].”). SL.4.4 Report on a topic or text, tell a story, or recount an experience in an organized manner, using appropriate facts and relevant, descriptive details to support main ideas or themes; speak clearly at an understandable pace.
Vocabulary:	- **Career** - A job you work toward by building skills and learning new things. - **Computer Science** - Using computers to solve problems and create things. - **Event** - An action that causes something to happen. - **Sprite** - A graphic on the screen with a location, size, and appearance.
Notes:	

Week 10: Code.org, Course E, Lesson 6 – Game Design

Lesson overview:	**Purpose:** This lesson is intended to prepare students to create simple game projects in Sprite Lab, a block-based programming environment where you can make simple animations and games with objects and characters that interact with each other. Students will begin to form an understanding of the programming model of this tool, and explore ways they can use it to express themselves. In this skill-building lesson, students will work through a series of programming levels on the computer, finishing with an open-ended "free play" task where they can build whatever they like. Students will write programs and learn about the two concepts at the heart of Sprite Lab: sprites and behaviors. Students will learn to create and animate sprites and make them interactive using events. **Lesson:** ● Introduction – Use key vocabulary slide and lesson objective slide. ● Sprite Review – Say: Sprites are objects that have different attributes or properties. The object is like a car that has different properties like size, color, make, model, and year. Or it's like a dog that has a size, breed, age, color. Each sprite can be a little different and you can set the properties with code. ● Code Review – Read the code one block at a time starting at the top. Point to each block and each image as you go. Make a note about sequence and how we couldn't put the behavior block before the make sprite block because the computer reads code form top to bottom accentuating the remark above. **Discuss:** What is the same or different about our instructions and this code? ● Game Design Setup – show video Students will work at their own pace. ● Sprite Behaviors ● Events – show video ● Reflection – Students will write a description of their sprite using adjectives that follow typical order patterns such as opinion, size, age, shape, color, origin, material, and purpose. Students could take a photo of their sprite, place it in a google slide, and add their written description. Students can share these slides in a whole group setting. ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	Course E, Lesson 6 – Game Design
CS standards addressed:	The students will be able to: ● Create groups of sprites, and control a single sprite from the group using events. ● Create new sprites. Standards:

	- **1A-AP-11** - Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. - **1B-AP-10** - Create programs that include sequences, events, loops, and conditionals.
Time needed:	**Total Time: 60 min** - Warm Up/Introduction **5 min** - Main Activity **35 min** - Wrap Up/Reflection **5 min** - Keyboarding Practice **15 min**
Materials needed:	Teachers: - Smartboard/projector with sound - **CSF - Course E - Slides** - Slides (**Download**) - **Helpful Hints** - Resource (**Download**) - **Intro to Sprite Lab** - Video Students: - Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	L.4.1d Demonstrate command of the conventions of standard English grammar and usage when writing or speaking. d. Order adjectives within sentences according to conventional patterns (e.g., a small red bag rather than a red small bag).
Vocabulary:	- **Behavior** - An action that a sprite performs continuously until it's told to stop. - **Event** - An action that causes something to happen. - **Sprite** - A graphic on the screen with a location, size, and appearance.
Notes:	

Week 11: Code.org, Course E, Lesson 7 – Variables as Score

Lesson overview:	**Purpose:** In this skill-building lesson students will use variables to track a value that changes over time, like a counter or a scoreboard in a game. Students will begin learning how to modify the data stored in a variable by setting the initial value in the program and using events that cause the data to change based on user interaction. This key understanding of how variables work will also enable them to create more advanced projects and games. **Lesson:** ● Introduction ○ Using the slides provided display and discuss key vocabulary and objectives of lesson. ● Code Review and Discussion ○ Display the code and have students read each line. ● Variables as Score – display and discuss slides 4-10 ● Skill Building ● Winning Conditions – display and discuss slides 8-10 (Remind students to write the conditional statements as complete sentences.) Model writing a complete sentence bfore students start writing the conditions. ● Debugging Practice – Students will practice and have an opportunity of free play. ● Reflection ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	Course E Lesson 7 : Variables as Score
CS standards addressed:	Students will be able to: ● Use a conditional statement to check for a winning score ● Use a variable that stores information and changes over time Standards: ● **1A-AP-11** - Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. ● **1B-AP-09** - Create programs that use variables to store and modify data. ● **1B-AP-10** - Create programs that include sequences, events, loops, and conditionals.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **5 min** ● Main Activity **35 min** ● Wrap Up/Reflection **5 min** ● Keyboarding Practice 1**5 min**

Materials needed:	Teachers: - Smartboard/projector with sound Students: - Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	L.4.1f Demonstrate command of the conventions of standard English grammar and usage when writing or speaking. f. Produce complete sentences, recognizing and correcting inappropriate fragments and run-ons.
Vocabulary:	Condition - A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored. Conditionals - Statements that only run under certain conditions. Variable - Variables store information (data). The value stored in a variable can be updated throughout a program.
Notes:	

Week 12: Code.org, Course E, Lesson 8—Environment and Players

Lesson overview:	**Purpose:** In this skill-building lesson, students will delve into the world of game mechanics while furthering their understanding of Sprite Lab. Students will explore the mechanics that make games fun and engaging. They will learn about concepts such as player movement, obstacles, scoring systems, and win/lose conditions. They will begin to assign roles to different elements within their games learning to differentiate between sprites that act as players (controlled by the user) and sprites that form the environment (static elements or obstacles). They will understand the roles these sprites play in shaping gameplay dynamics. Throughout the lesson, students will apply their knowledge by designing and implementing game mechanics using Sprite Lab. By the end of the lesson, they will have a solid grasp of how to create interactive and engaging game experiences within the framework of Sprite Lab and will be ready to code their own creative game in the following lesson! **Lesson:** ● Introduction ○ Display and discuss vocabulary and lesson objectives using the slides provided. ○ Use the slides to discuss the 3 examples provided ○ Environment Type – Students can explore a level that interests them. ○ Player Type – student will practice. ○ CFU and Practice – students will match the code on the right with images on the left. ● Reflection ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	**Course E, Lesson 8—Environment and Players**
CS standards addressed:	Students will be able to: ● Assign sprites as player type or environment type ● Explore game mechanics Standards: ● **1A-AP-11** - Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. ● **1B-AP-09** - Create programs that use variables to store and modify data.

	- **1B-AP-10** - Create programs that include sequences, events, loops, and conditionals.
Time needed:	**Total Time: 60 min** - Warm Up/Introduction **10 min** - Main Activity **35 min** - Wrap Up/Reflection **5 min** - Keyboarding Practice **10 min**
Materials needed:	Teachers: - Smartboard/projector with sound - **CSF - Course E - Slides** - Slides (**Download**) - **Helpful Hints** - Resource (**Download**) Students: - Student devices with access to the internet
Subject integrated:	Social Studies
Other standards addressed:	G.4.3.2. Distinguish between cardinal and intermediate directions (e.g., north, northeast, northwest, southeast, southwest, east, and west).
Vocabulary:	**Condition** - A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored. **Conditionals** - Statements that only run under certain conditions. **Game Mechanics** - Define how the game works and what determines success or failure in the game. **Sprite Type** - Identifies sprites as player or environment, which determines how they interact with sprites of other types. **Variable** - Variables store information (data). The value stored in a variable can be updated throughout a program.
Notes:	

Week 13: Code.org, Course E, Lesson 9 – Game Jam Day 1: Create

Lesson overview:	**Purpose:** **Get ready to join a Game Jam!** In this creative project-based lesson, students will apply their newfound skills in Sprite Lab to develop an original game from start to finish in the context of game development industry practice: Game Jam. Students will experience the thrill of rapid game development as they participate in a collaborative challenge to create a new game based on a randomly generated theme. Working against the clock, they'll brainstorm, plan, and code their games, embracing the creative constraints of the theme to produce innovative and engaging gameplay experiences. Students will include variables in their games to track and modify values dynamically. By the end of the lesson, they will have gained experience in game design, coding, and problem-solving, ready to share their creations with others in a user testing experience in the next lesson. **Lesson:** ● Introduction ● Watch these videos for reference: The first video is a thorough explanation of how game jams work. The second video features students in Bulgaria who have participated in a Game Jam and are sharing their thoughts. Note that the second video features subtitles in English, which you may want to read aloud for students. ● Game Jam Set Up – There is a slide to use for students who might need help thinking of ideas for their selected game mechanic. ● Make the Game - Students will work together to code their game. Relate this to the engineering design process in Science. ● Reflection
Lesson links/resources:	Course E Lesson 9: Game Jam Day 1: Create
CS standards addressed:	Students will be able to: • Participate in a Game Jam to create a new game based on a randomly generated theme. • Program a variable that changes throughout the game. • Utilize new Sprite Lab & coding skills to develop (plan and code) a creative game. Standards • **1A-AP-11** - Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. • **1B-AP-09** - Create programs that use variables to store and modify data.

	- **1B-AP-10** - Create programs that include sequences, events, loops, and conditionals.
Time needed:	**Total Time: 60 min** - Warm Up/Introduction **15 min** - Main Activity **35 min** - Wrap Up/Reflection **10 min**
Materials needed:	Teachers: - Smartboard/projector with sound - **CSF - Course E - Slides** - Slides (**Download**) - **Helpful Hints** - Resource (**Download**) - **[KEY] Game Plan Activity Guide** - Answer Key (**Download**) Students: - Student devices with access to internet
Subject integrated:	Science
Other standards addressed:	P.4.6A.6 Design a device that converts any form of energy from one form to another form (e.g., construct a musical instrument that will convert vibrations to sound by controlling varying pitches, a solar oven that will convert energy from the sun to heat energy, or a simple circuit that can be used to complete a task). Use an engineering design process to define the problem, design, construct, evaluate, and improve the device.*
Vocabulary:	**Game Jam** - A creative event where people come together in teams to make video games based on a chosen theme within a limited amount of time. **Game Mechanics** - Define how the game works and what determines success or failure in the game. **Variable** - Variables store information (data). The value stored in a variable can be updated throughout a program.
Notes:	

Week 14: Code.org, Course E, Lesson 10—Game Jam Day 2: Play

Lesson overview:	**Purpose:** In this culminating lesson, students will learn how to collect and analyze valuable data from users of their games. This will empower them to make informed decisions for future iterations and improvements. Through hands-on activities and discussions, students will delve into the importance of user feedback and data-driven decision-making in game development.By the end of the lesson, they'll understand how to leverage the data they collected in order to iterate on their games, refine gameplay mechanics, user interfaces, and the overall player experience. This final lesson equips students with the skills and mindset needed to create games that continuously evolve and improve based on user insights. **Lesson:** ● Introduction ○ Review vocabulary and lesson objectives using the slides provided. ● User Testing – Play the game using the slide ● Data Analysis – students will compile the feedback into graphs so it is easier to see the patterns. ● Reflection ○ Share out ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	Course E Lesson 10: Game Jam Day 2: Play
CS standards addressed:	Students will be able to: ● Analyze data from users in order to determine areas for improvement. ● Collect useful data from users of your game. Standards: ● **1B-AP-13** - Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. ● **1B-DA-06** - Organize and present collected data visually to highlight relationships and support a claim. ● **1B-DA-07** - Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.

Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **5 min** ● Main Activity **35 min** ● Wrap Up **5 min** ● Keyboard Practice **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound ● **CSF - Course E - Slides** - Slides (**Download**) ● **Helpful Hints** - Resource (**Download**) ● **[KEY] User Data Activity Guide** - Answer Key (**Download**) Students: ● Student devices with access to the internet ● **User Data Activity Guide** - Activity Guide (**Download**)
Subject integrated:	ELA
Other standards addressed:	L.4.6 Acquire and use accurately grade-appropriate general academic and domain specific words and phrases, including those that signal precise actions, emotions, or states of being (e.g., quizzed, whined, stammered) and that are basic to a particular topic (e.g., wildlife, conservation, and endangered when discussing animal preservation).
Vocabulary:	**Data** - Collection of information for reference or analysis. **User testing** - Asking real people to try out a product and then make it better based on user feedback.
Notes:	

Week 15: Interactive Presentation

Lesson overview:	**Purpose:** Students will create an interactive presentation to explain the properties of heat, electrical, sound, and light energy. **Lesson:** ● Students watch videos and create a "Dialogue" project in Scratch Account. ● Videos: Introduction to Scratch. (Use smartboard to show videos in attached slide or assign the slide to students in google classroom for them to view independently.) *Youtube has some additional instructional videos if needed. ● Demonstrate on smart board how to open project in Scratch. ● Using *Interactive Presentation*, students will create a presentation (or use an existing one) and make it interactive using Scratch. Students add at least three slides, including a title slide that introduces their topic. Use this lesson to teach students how to present reports, research, or write alternative story endings in a way that engages an audience and encourages collaboration. ● This lesson was designed for students in Grades 3 through 5 and can be adapted for many different ages and audiences. It takes approximately an hour to run.
Lesson links/resources:	Interactive Presentation Lesson Plan Scratch Starter Project
CS standards addressed:	Students will be able to: ● Create an interactive presentation using Scratch Standards ● **AP.1B.4**—Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. ● **AP.1B.6**—Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **7 min** ● Main Activity **45 min** ● Wrap Up/Reflection **8 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to internet
Subject integrated:	Science
Other standards addressed:	● **P.4.6A**—Students will demonstrate an understanding of the common sources and uses of heat and electric energy and the materials used to transfer heat and electricity. ● **P.4.6B**—Students will demonstrate an understanding of the properties of light as forms of energy.

	• **P.4.6C**—Students will demonstrate an understanding of the properties of sound as a form of energy.
Vocabulary:	Parallelism: The process of events happening at the same time, either independently or interdependently Debugging: The process of identifying and fixing error(s) in a program when it is not functioning as expected Control structures: Sections of code that order the direction or flow of how a program functions. The control structure in this lesson is focused on loops.
Notes:	→ Teachers will need to create FREE teacher and/or student accounts https://scratch.mit.edu/join A teacher account does not have to be created to use Scratch. However, if you and your students establish a free account, your students can share their creations with you and other students. Also, students will not be able to save their work online without an account. This lesson was recreated using resources from CS First which is no longer available online.

Week 16: Code.org, Course E, Lesson 11—Songwriting

Lesson overview: Unplugged	**Purpose:** The use of functions helps simplify code and develop the student's ability to organize their program. Students will quickly recognize that writing functions can make their long programs easier to read and easier to debug if something goes wrong. **Lesson:** ● Introduction ○ Key vocabulary ○ Sing a song ● Songwriting ○ Songwriting with Functions - video ○ Functions Unplugged: Songwriting - handout ● Assessment ○ Functions Unplugged: Songwriting - assessment ● Reflection ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	● Course E Lesson 11- Songwriting ● The Mississippi State Song "Go, Mississippi" Lyrics ● State Song of Mississippi- Go, Mississippi
CS standards addressed:	Students will be able to: ● Describe how functions can make programs easier to write ● Identify sections of a song to pull into a function ● Locate repeating phrases inside song lyrics Standards ● **AP.1B.1**—Compare and refine multiple algorithms for the same task and determine which is the most appropriate. ● **AP.1B.1a**—Students should be able to look at different ways to solve the same task and decide which would be the best solution. ● **AP.1B.4**—Decompose (break down) problems into smaller, manageable subproblems to facilitate the program. ● **AP.1B.4a**—Students should be able to break down problems into smaller, manageable tasks.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **20 min** ● Main Activity **20 min** ● Wrap Up **5 min** ● Keyboarding Practice **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound ● Functions Unplugged: Songwriting-Lesson in action video ● Functions Unplugged: Songwriting-Assessment answer key for the students ● Songwriting with Functions-Unplugged video (Download) Students:

	- Functions Unplugged: Songwriting-Worksheet - Functions Unplugged: Songwriting-Assessment
Subject integrated:	Social Studies
Other standards addressed:	**H.4.1**—Recognize symbols, customs, and celebrations representative of our community, Mississippi, and the United States.
Vocabulary:	Function: A piece of code that you can easily call over and over again
Notes:	

Week 17: Sequential Order Using Scratch

Lesson overview:	**Purpose:** The students will build skills essential to reading comprehension by retelling a story through a plugged coding activity using Scratch. **Lesson:** ● The student will create an animation that recalls the details and events of the story chosen by the teacher, in sequential order. ● Students can use the *Create a Story Tutorial* to create their story. ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	Scratch: Story Guide Scratch (Click Tutorials and Create a Story for a template)
CS standards addressed:	Students will be able to: ● Build story retelling skills through the creation of an animation. Standards: ● **AP.1B.3—**Create programs that include sequences, events, loops, and conditionals.
Time needed:	**Total Time: 60 min** ● Expectations/Directions **10 min** ● Creating Animation **40 min** ● Keyboarding **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	● **RL.4.1**—Refer to details and examples in a text when explaining what the text says explicitly and when drawing inferences from the text. ● **RI.4.1**—Refer to details and examples in a text when explaining what the text says explicitly and when drawing inferences from the text.
Vocabulary:	Behavior: An action that a sprite performs continuously until it is told to stop Algorithm: A list of steps to finish a task

Notes:	→ Teachers will need to create FREE teacher and/or student accounts https://scratch.mit.edu/join A teacher account does not have to be created to use Scratch. However, if you and your students establish a free account, your students can share their creations with you and other students. Also, students will not be able to save their work online without an account. This lesson was recreated using resources from CS First which is no longer available online.

Week 18: Code.org, Course E, Lesson 12—Functions in Music Lab

Lesson overview:	**Purpose:** In this lesson, students will act as music producers, refining remixes in Music Lab while learning how functions can simplify and organize their code. They will learn how to make a new function and then call it into the main program. This lesson demonstrates how producers combine technology and creativity with a song's structure to create polished tracks. **Lesson:** ● Introduction ○ Key vocabulary (Use slides) ○ Discuss: What are the benefits of efficient code? ● Dance Moves ● A dance is like a function ● Activity – Watch the video first and students can work independently or in pairs. ● Reflection – Remind the students of the opening video from a real producer. Students will research famous Mississippians known for their artwork, music, architecture, and literature. Students will create a presentation to share with peers. ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	● Course E Lesson 12: Functions in Music Lab
CS standards addressed:	Students will be able to: ● Customize and call functions that organize and simplify your code. Standards: **1B-AP-08** - Compare and refine multiple algorithms for the same task and determine which is the most appropriate. **1B-AP-10** - Create programs that include sequences, events, loops, and conditionals. **1B-AP-11** - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. **1B-AP-15** - Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **20 min** ● Main Activity **20 min** ● Wrap Up **5 min** ● Keyboarding Practice **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound ● **Lesson Slides** - Slides (**Download**) Students:

	- Computer and internet access
Subject integrated:	Social Studies
Other standards addressed:	H.4.4.1Identify Mississippians known for their artwork, music, architecture, and literature.
Vocabulary	**Efficient** - Doing something in the easiest and quickest way. **Function** - A piece of code that you can call over and over again.
Notes:	

Week 19: Research a Scientist Project

Lesson overview: Coding	**Purpose:** Students will research scientists who pioneered in the science of sound and create a presentation to share with the class. **Lesson:** ● Introduction ○ Choose a scientist ● Code Your Scientist ○ Show "Code Your Hero" video (the video refers to "hero", but remind students their project will be about a scientist). ○ Students create their animations in Scratch ● Reflection
Lesson links/resources:	Research a Scientist Project - the lesson plan refers to "hero", but remind students their project will be about a scientist.
CS standards addressed:	Students will be able to ● Use event blocks (like "when flag clicked") to trigger a series of code ● Sequence "say" and "wait" blocks to make their hero speak and have a dialogue with another character ● Program actions to happen using "when key pressed" events. ● Move an object across the screen using motion blocks ● Repeat actions using loop blocks ● Program their hero to score points when they perform a certain action ● Draw their own hero using the "Paint Editor" in the *Scratch for CS First* coding editor. (This add-on video pairs well with the Optional Planning Activity.) Standards ● **AP.1B.2**—Create programs that use variables to store and modify data. ● **AP.1B.3**—Create programs that include sequences, events, loops, and conditionals.
Time needed:	**Total Time: 60 min** ● Introduction **3 min** ● Main Activity **50 min** ● Wrap Up **7 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject(s) integrated:	Science Writing
Other standards addressed:	Science ● **P.4.6C.3**—Obtain and communicate information about scientists who pioneered in the science of sound, (e.g., Alexander Graham Bell, Robert Boyle, Daniel Bernoulli, and Guglielmo Marconi). Writing

	- **W.4.7**—Conduct short research projects that build knowledge through investigation.
Vocabulary:	Parallelism: The process of events occurring simultaneously, either independently or interdependently Debugging: The process of identifying and fixing errors in a program that is not functioning as expected Control structure: Sections of code that order the direction or flow of how a program functions. The control structure in this lesson is focused on loops Variable: A container that stores a value that can change
Notes:	

Week 20: Civil Rights History

Lesson overview:	**Purpose:** Students will talk about an important event in Civil Rights History by creating a story in Scratch. The students must choose a background and use two sprite characters in their animation. **Lesson:** ● Introduction and expectation of assignment ● Scratch (Students can use the Create a Story Tutorial to create their Civil Rights historical event) ○ Students choose a background ○ Students choose the sprite characters for their animation ○ Animations should be about important Civil Rights history and/or events ○ See CS standards for components students should use in their projects (sequences, events, loops, and conditions). ● Reflection/Share
Lesson links/resources:	Scratch: Story Guide Scratch
CS standards addressed:	Students will be able to: ● Use event blocks (like "when flag clicked") to trigger a series of code ● Sequence at least 3 "say" blocks between two sprites (characters) to construct a dialogue ● Program a conditional so that the computer can decide based on a user response ● Produce repeated movements by applying control blocks to their program Standard ● **AP.1B.3a**—Students should be able to create programs that include sequences, events, loops, and conditions.
Time needed:	**Total Time: 60 min** ● Introduction **5 min** ● Civil Rights Project **45 min** ● Reflection/Share **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	Social Studies
Other standards addressed:	**4.MS.8**—Analyze the Civil Rights Movement to determine the social, political, and economic impact on Mississippi.

Vocabulary:	Control structures: Sections of code that order the direction of flow of how a program function. Control structures include conditionals and loops. Debugging: The process of identifying and fixing error(s) in a program when it is not functioning as expected
Notes:	

Week 21: Animating the Steps of Long Division

Lesson overview:	**Purpose:** The goal of this lesson is to utilize facets of coding to facilitate accuracy and skill-building within the multi-step process of long division. Students will use block-coding to animate the mnemonic device "Does McDonald's Sell Cheeseburgers?" which is used to help them remember the many steps of long division. **Lesson:** ● Introduction ○ Review the steps of long division ○ Watch the "Animate a Name" tutorial on Scratch ● Word Animation ○ Pick a background ○ Select the desired letters ○ Code sprite characters ○ Run the code ● Assessment ○ Student presentations ● Wrap Up
Lesson links/resources:	Scratch
CS standards addressed:	Students will be able to: ● Look at different ways to solve the same task and decide which would be the best solution ● Break down problems into smaller, simpler tasks Standards: ● **AP.1B.1**—Compare and refine multiple algorithms for the same task and determine which is the most appropriate. more advanced features. ● **AP.1B.4**—Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
Time needed:	**Total Time: 60 min** ● Warm Up/Bridging **10 min** ● Main Activity **40 min** ● Wrap Up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	Math
Other standards addressed:	**4.NBT.6**—Find whole-number quotients and remainders with up to four-digit dividends and one-digit divisors, using strategies based on place value, the properties of operations, and/or the relationship between multiplication and division. Illustrate and explain the calculation by using equations, rectangular arrays, and/or area models.

Vocabulary:	Algorithm: A list of steps to finish a task Behavior: An action that a sprite performs continuously until it is told to stop Parallelism: The process of events occurring simultaneously, either independently or interdependently Variable: A container that stores a value that can change
Notes:	

Week 22: Theme, Main Idea, and Details

Lesson overview:	Purpose: The students will build skills essential to reading comprehension. The student will determine a theme of a story, drama, or poem chosen by the teacher, or determine the main idea of a text and explain how it is supported by key details through a plugged-in coding activity (animation). Lesson: ● Introduction ○ Recall theme, main idea, and details ○ Introducing lesson activity. TTW assign a short, story, drama, or poem ○ Provide instructions on what to include in animation ● Pre-Writing ○ Brainstorming ● Create Animation ● Wrap Up ○ Reflection/Share
Lesson links/resources:	Scratch: Story Guide Scratch
CS standards addressed:	Students will be able to: ● Determine the theme or main idea of a story, drama, or poem, and explain how it is supported, through the creation of an animation Standards: ● **AP.1B.10a**—Students should explain code choices using comments within the code, presentations, and demonstrations. ● **1B.AP.10**—Create programs that include sequences, events, loops, and conditionals.
Time needed:	**Total Time: 60 min** ● Introduction **10 min** ● Main Activity **40 min** ● Wrap Up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	● **RL.4.2**—Determine the theme of a story, drama, or poem from details in the text. ● **RI.4.2**—Determine the main idea of a text and explain how it is supported by the key details.
Vocabulary:	Behavior: An action that a sprite performs continuously until it is told to stop Algorithm: A list of steps to finish a task

Week 23: Code.org, Course E, Lesson 13—Functions in Minecraft

Lesson overview: Coding	**Purpose:** Students will discover the versatility of programming by practicing functions in different environments. Here, students will recognize reusable patterns and be able to incorporate named blocks to call predefined functions. **Lesson:** ● Introduction ○ Key vocabulary ● Bridging Activity (choose one) ○ Unplugged activity using some blocks ○ Preview of online puzzles ● Functions in Minecraft ○ Skill building ○ Minecraft-repeat loops (video) ○ Skill building ○ Minecraft-functions (video) ○ Skill building ○ Minecraft-congratulations (video) ○ Free play ● Reflection
Lesson links/resources:	Course E Lesson 13: Functions in Minecraft Unplugged Blocks (Courses C-F)
CS standards addressed:	Students will be able to: ● Use functions to simplify complex programs ● Use predetermined functions to complete commonly repeated tasks Standards: ● **1B-AP-08** - Compare and refine multiple algorithms for the same task and determine which is the most appropriate. ● **1B-AP-10** - Create programs that include sequences, events, loops, and conditionals. ● **1B-AP-11** - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. ● **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. ● **1B-AP-15** - Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
Time needed:	**Total Time: 60 min** ● Warm Up **10 min** ● Bridging Activity **15 min** ● Main Activity **30 min** ● Wrap Up/Reflection **5 min**
Materials needed:	Teachers: ● Smartboard/projector with sound

	Students: - Student devices with access to the internet
Subject integrated:	Math
Other standards addressed:	**4.OA.5**—Generate a number or shape pattern that follows a given rule. Identify apparent features of the pattern that were not explicit in the rule itself.
Vocabulary:	Function: A piece of code that you can easily call over and over again
Notes:	

Week 24: Life Cycles

Lesson overview:	Purpose: Students will create an interactive presentation to explain the life cycle of a specific plant or animal. Lesson: ● Introduction: Review various life cycles ● Create an animation showing the steps in chosen life cycle ● Play the animation showing the cycle repeating in a loop
Lesson links/resources:	Scratch
CS standards addressed:	Students will be able to: ● Create an animation showing the life cycle of a chosen animal Standards: ● **AP.1B.10a**—Students should explain code choices using comments within the code, presentations, and demonstrations. ● **1B.AP.10**—Create programs that include sequences, events, loops, and conditionals.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **10 min** ● Main Activity **35 min** ● Wrap Up **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	Science
Other standards addressed:	**L.4.2**—Students will demonstrate an understanding of life cycles, including familiar plants and animals (e.g., reptiles, amphibians, or birds).
Vocabulary:	Loop**:** The action of doing something over and over again Repeat**:** To do something again
Notes:	

<table>
<tr><th colspan="2"><u><b>Week 25: Code Your Hero</b></u></th></tr>
<tr><td>Lesson overview:</td><td><u>Purpose:</u>
The teacher will tell the students that they are going to bring their coding to life. They will create an interactive activity by turning a famous person that they look up to into a superhero.
<u>Lesson:</u>
● Introduction:
○ Discuss the definition of hero (will likely mean different things to different people)
○ Allow students a few minutes to think of a person they look up to.
● Students will watch videos and create a project in Scratch.
● Videos: Introduction to Scratch. (Use smartboard to show videos in attached slide or assign the slide to students in google classroom for them to view independently.) *Youtube has some additional instructional videos if needed.
● Demonstrate on smart board how to open the project in Scratch.

● Main Activity
○ Choose your sprite
○ Choose your background/setting
○ Code your animation to discuss your famous superhero
● Reflection/Share
● Keyboarding Practice
○ If time remains, have students use resources from above to practice keyboarding.</td></tr>
<tr><td>Lesson links/resources:</td><td>Code Your Hero Lesson Plan Slide</td></tr>
<tr><td>CS standards addressed:</td><td>Students will be able to:
● Create/present a program that includes sequences, loops, events, and conditionals, in the form of an animation
Standards:
● <b>AP.1B.3a</b> Students should be able to create programs that include sequences, events, loops, and conditionals.
● <b>AP.1B.6b</b> Students should document the plan as, for example, a storyboard, flowchart, pseudocode, or story map.</td></tr>
<tr><td>Time needed:</td><td><u><b>Total Time:</b></u> <b>60 min</b>
● Introduction <b>5 min</b>
● Main Activity <b>40 min</b>
● Wrap Up <b>5 min</b>
● Keyboarding <b>10 min</b></td></tr>
<tr><td>Materials needed:</td><td>Teachers:
● Smartboard/projector with sound
Students:
● Student devices with access to the internet</td></tr>
<tr><td>Subject integrated:</td><td>ELA</td></tr>
</table>

Other standards addressed:	**W.4.1**—Write opinion pieces on topics or texts, supporting a point of view with reasons and information.
Vocabulary:	Debugging: The process of identifying and fixing errors in a program that is not functioning as expected Control structures: Sections of code that order the direction of flow of how a program function. Control structures include conditionals and loops Variable: A container that stores a value that can change
Notes:	→ Teachers will need to create FREE teacher and/or student accounts https://scratch.mit.edu/join A teacher account does not have to be created to use Scratch. However, if you and your students establish a free account, your students can share their creations with you and other students. Also, students will not be able to save their work online without an account. This lesson was recreated using resources from CS First which is no longer available online.

Week 26: Subtracting Across Zeros

Lesson overview: Coding	**Purpose:** The goal of this lesson is to utilize facets of coding to facilitate accuracy and skill-building within the multi-step process of subtracting across zeros. Students will use block-coding to animate their knowledge of the steps to accurately subtract across zeros to derive the difference of two numbers. **Lesson:** ● Introduction ○ Recall steps of subtracting across zeros ○ Discuss remixing ● Activity (Remixing) ○ Background ○ Sprite ○ Animations ■ Explain how to subtract across zeros within ○ Run the code ● Assessment ○ Students "teaching" with their animations ● Reflection/Share
Lesson links/resources:	Scratch: School-Place Value (there are other examples of remixes in the bottom right-hand corner)
CS standards addressed:	Students will be able to: ● Modify and/or reuse portions of an existing program into their own work to create something new ● Work collaboratively to compare and refine multiple algorithms to produce a cohesive, effective presentation Standards: ● **AP.1B.1**—Compare and refine multiple algorithms for the same task and determine which is the most appropriate. ● **AP.1B.5**—Modify, remix, or incorporate portions of an existing program into one's own work to develop something new or add more advanced features.
Time needed:	**Total Time: 60 min** ● Introduction **5 min** ● Activity **35 min** ● Assessment **10 min** ● Reflection/Share **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	Math
Other standards addressed:	**4.NBT.1**—Explain that in a multi-digit whole number, a digit in one place represents 10 times as much as it represents in the place to its right, up to 100,000. **4.NBT.4**—Fluently add and subtract (including subtracting across zeros) multi-digit whole numbers using the standard algorithm.

Vocabulary:	Program: An algorithm that has been coded into something that can be run by a machine Programming: The art of creating a program
Notes:	

Week 27: Literal and Figurative Language

Lesson overview:	Purpose: This lesson plan is designed to help you teach figurative language in a fun, visual, and engaging way through coding. Lesson: ● Introduction ○ Review figurative language: Metaphors, similes, personification, hyperbole, and idioms are just some types of figurative language. ● Show the *Figurative Language-Example Project (Simile)* ● Pre-writing (Brainstorming) ● Students watch videos and create a project in Scratch. ● Videos: Introduction to Scratch. (Use smartboard to show videos in attached slide or assign the slide to students in google classroom for them to view independently.) *Youtube has some additional instructional videos if needed. ● Demonstrate on smart board how to open the project in Scratch. ● Create Animation
Lesson links/resources:	● Lesson Plan Slide ● Figurative Language **Lesson Plan** ● Scratch: Figurative Language Examples
CS standards addressed:	Students will be able to: ● Create programs that include sequences, events, loops, and conditionals ● Decompose problems into smaller problems Standards: ● **AP.1B.10** Create programs that include sequences, events, loops, and conditionals. ● **AP.1B.11** Decompose problems into smaller, manageable subproblems to facilitate the program development process.
Time needed:	**Total Time: 60 min** ● Introduction **10 min** ● Main Activity **40 min** ● Wrap Up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	**RL.4.4**—Determine the meaning of words and phrases as they are used in a text, including those that allude to significant characters found in mythology (e.g., Herculean).

Vocabulary:	Parallelism: The process of events happening at the same time, either independently or interdependently Debugging: The process of identifying and fixing error(s) in a program when it is not functioning as expected Control structures: Sections of code that order the direction or flow of how a program functions. The control structure in this lesson focuses on loops.
Notes:	This lesson was recreated using resources from CS First which is no longer available online.

Week 28: Code.org, Course E, Lesson 14—Functions With Artist

Lesson overview:	**Purpose:** One of the most important components to this lesson is providing students with a space to create something they are proud of. These puzzles progress to more and more complex images, but each new puzzle only builds off the previous puzzle. At the end of this lesson, students will feel confident and proud of their hard work. **Lesson:** ● Introduction ○ Recall the use of a function ● Functions with Artist ○ Prediction ○ Skill building ○ Challenge ○ Practice ○ Lesson extras ● Reflection ○ Students will share what they felt the hardest and easiest parts of the lesson were
Lesson links/resources:	Course E Lesson 14: Functions With Artist
CS standards addressed:	Students will be able to: ● Categorize and generalize code into useful functions ● Recognize when a function could help to simplify a program Standards: ● **1B-AP-08** - Compare and refine multiple algorithms for the same task and determine which is the most appropriate. ● **1B-AP-10** - Create programs that include sequences, events, loops, and conditionals. ● **1B-AP-11** - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. ● **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. ● **1B-AP-17** - Describe choices made during program development using code comments, presentations, and demonstrations.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **15 min** ● Main Activity **30 min** ● Wrap Up/Reflection **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound

	- **CSF - Course E - Slides** - Slides (**Download**) Students: - Student devices with access to the internet - **Unplugged Blocks (Courses C-F)** - Manipulatives
Subject integrated:	Math
Other standards addressed:	**4. MD.5**—Recognize angles as geometric shapes that are formed wherever two rays share a common endpoint and understand concepts of angle measurement.
Vocabulary:	Function: A piece of code that you can easily call over and over again
Notes:	

Week 29: Code.org, Course E, Lesson 15—Conditionals in Minecraft: Voyage Aquatic

Lesson overview:	**Purpose:** This set of puzzles will work to solidify and build on the knowledge of loops and introduce conditionals. By pairing these two concepts together, students will be able to explore the potential for creating fun and innovative programs in a new and exciting environment. **Lesson:** ● Introduction ○ Key vocabulary ● Conditionals in Minecraft: Voyage Aquatic ○ Minecraft: Voyage Aquatic Introduction (video) ○ Skill Building ○ Minecraft: Voyage Aquatic Repeat Until (video) ○ Skill Building ○ Minecraft: Voyage Aquatic Conditionals (video) ○ Skill building ○ Minecraft: Voyage Aquatic Congratulations ○ Free play ● Reflections
Lesson links/resources:	Course E Lesson 15: Conditionals in Minecraft: Voyage Aquatic
CS standards addressed:	Students will be able to: ● Define circumstances when certain parts of a program should run and when they should not ● Determine whether a conditional is met based on criteria Standards ● **AP.1B.3**—Create programs that include sequences, events, loops, and conditionals.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **15 min** ● Main Activity **30 min** ● Wrap Up/Reflection **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	Math
Other standards addressed:	**4.NBT.5—**Multiply a whole number of up to four digits by a one-digit whole number, and multiply two two-digit numbers, using strategies based on place value and the properties of operations. Illustrate and explain the calculation by using equations, rectangular arrays, and/or area models.
Vocabulary:	Condition: Something a program checks to see if it is true before allowing an action Conditionals: Statements that only run under certain conditions

Week 30: Weather Safety	
Lesson overview:	**Purpose:** Students will create an animated story with a stormy setting to discuss weather safety. **Lesson:** ● Introduction ○ Review weather terms ● Main Activity ○ Students will follow along with the storytelling videos while working to create their own animated story using scratch. ○ Note: The slide has a series of lesson on storytelling lessons. You can make the project as detailed as you wish. The lesson can be extended to other days if needed. ● Wrap Up ○ Students will share their stories with a partner ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	● Storytelling Lesson Plan Slide
CS standards addressed:	Students will be able to: ● Students will create an animated story with a story setting to discuss weather safety using Scratch Standards: ● **AP.1B.10—**Create programs that include sequences, events, loops, and conditionals. ● **AP.1B.11—**Decompose problems into smaller, manageable subproblems to facilitate the program development process.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **5 min** ● Main Activity **35 min** ● Wrap Up **10 min** ● Keyboarding Practice **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	Science
Other standards addressed:	**E.4.9C.5—**Obtain and communicate information about severe weather phenomena (e.g., thunderstorms, hurricanes, or tornadoes) to explain steps humans can take to reduce the impact of severe weather events
Vocabulary:	Condition: Something a program checks to see if it is true before allowing an action Conditionals: Statements that only run under certain conditions While loop: A loop that continues to repeat while a condition is true

Notes:	This lesson was recreated using resources from CS First which is no longer available online.

Week 31: Famous Mississippians

Lesson overview:	Purpose: Students will choose a famous Mississippian and write a story about their accomplishments. They will make this story come to life by utilizing Scratch. Lesson: ● Introduction ○ Project expectations ○ Choose your famous Mississippian ● Main Activity ○ Create animation ● Wrap Up and Share
Lesson links/resources:	Scratch: Story Guide Scratch
CS standards addressed:	Students will be able to: ● Create an animation Standards: ● **AP.1B.3a—**Students should be able to create programs that include sequences, events, loops, and conditionals. ● **AP.1B.6b—**Students should document the plan as, for example, a storyboard, flowchart, pseudocode, or story map.
Time needed:	**Total Time: 60 min** ● Introduction **10 min** ● Main Activity **40 min** ● Wrap Up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	Social Studies
Other standards addressed:	**CR.4.1.1—**Analyze the Civil Rights Movement to determine the social, political, and economic impact on Mississippi.
Vocabulary	Condition: Something a program checks to see if it is true before allowing an action Conditionals: Statements that only run under certain conditions While Loop: A loop that continues to repeat while a condition is true
Notes:	

Week 32: Points, Lines, Segments, Rays, and Angles

Lesson overview:	**Purpose:** *Robotics Lesson with Edison Robotics Robot* *If you do not have access to a robot (code & go mouse, Edison, Sphero Sprk+, etc.) a person/persons can "be" the robot* The student will remix a scratch lesson to review math lines, segments, rays, and angles. **Lesson:** ● Introduction ○ Review vocabulary terms for math ● Main Activity ○ In various groups, the student will remix Line Tracking Lesson to include different types of lines, such as (but not limited to) one set of parallel lines, one set of perpendicular lines, and two intersecting lines that are not perpendicular. ○ The student will then plug in their Edison Robot and upload their created maze to the Edison. ● Wrap Up ○ All groups will collaboratively reflect on the similarities and differences of the mazes created as well as the use of the various types of lines.
Lesson links/resources:	Scratch: Line Tracking
CS standards addressed:	Students will be able to: ● Remix a scratch lesson by adding different types of lines ● Upload created mazes to Edison Standards: ● **AP.1B.3—**Create programs that include sequences, events, loops, and conditionals.
Time needed:	**Total Time: 60 min** ● Introduction **10 min** ● Main Activity **40 min** ● Wrap Up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound ● Codable Robot (Some examples include: Edison robots, Code and Go Mouse, Botley, Dash) Students: ● Student devices with access to the internet
Subject integrated:	Math
Other standards addressed:	**4.G.1—**Draw points, lines, line segments, rays, angles (right, acute, obtuse), and perpendicular and parallel lines. Identify these in two-dimensional figures.

Vocabulary:	Program: An algorithm that has been coded into something that can be run by a machine Programming: The art of creating a program
Notes:	

Week 33: Digital Sharing Animation

Lesson overview:	**Purpose:** Students will apply their understanding of sharing personal and private information on the web by creating an interactive poster in this mini project, after reading and discussing 2 passages about digital sharing. The students will compare and contrast the information about the same topic in an animation. **Lesson:** ● Introduction ○ Recall information from previous digital sharing lessons about sharing personal and private information ● Main Activity ○ Pre-writing-brainstorming-Venn diagram ○ Create animation ● Wrap Up ○ Present interactive poster
Lesson links/resources:	Scratch
CS standards addressed:	Students will be able to: ● Use public domain and common media, and refrain from copying and using material created by others ● Discuss cybersecurity issues and protection of personal information Standards: ● **1B.IC.21—**Use public domain or creative commons media and refrain from copying or using material created by others without permission. ● **1B.NI.05—**Discuss real-world cybersecurity problems and how personal information can be protected.
Time needed:	**Total Time: 60 min** ● Introduction **10 min** ● Main Activity **40 min** ● Wrap Up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	**RI.4.9—**Integrate information from two texts about the same subject knowledgeably.
Vocabulary:	Copyright: The exclusive legal right to print, publish, perform, film, or record literary, artistic, or musical material, and to authorize others to do the same

Notes:

Week 34: Maze Mat Race

Lesson overview:	*Robotics Lesson with Code & Go Mouse*: Purpose: Reading comprehension lesson for students to strengthen their skills in referring to details, theme, character, setting, or word meaning, using interactive Maze Mat Race game. Lesson: ● Introduction ○ Follow prompts on the Maze Mat Race Lesson Plan ○ Discuss game rules and concept ● Main Activity ○ Divide the class into teams ○ Create a quiz using either Kahoot or Quizziz ○ Begin game ● Wrap Up ○ Debugging ○ Reflection
Lesson links/resources:	● Maze Mat Race Lesson Plan ● www.kahoot.com ● www.quizziz.com
CS standards addressed:	Students will be able to: ● Code a robot to maneuver through a maze Standards: ● **AP.1B.1—**Compare and refine multiple algorithms for the same task and determine which is the most appropriate. ● **AP.1B.4—**Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
Time needed:	**Total Time: 60 min** ● Introduction **10 min** ● Main Activity **40 min** ● Wrap Up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound ● Robot (Some examples include: Code and Go Mouse, Botley, Dash) ● Coding maze (Should be already coded before the game) ● Grade level passage (Can be passage previously discussed) ● Access to Kahoot or Quizizz Students: ● Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	● **RL.4.1—**Refer to details and examples in a text when explaining what the text says explicitly and when drawing inferences from the text. ● **RL.4.2—**Determine a theme of a story, drama, or poem from details in the text; summarize the text. ● **RL.4.3—**Describe in depth a character, setting, or event in a story

	or drama, drawing on specific details in the text (e.g., a character's thoughts, words, or actions). ● **RL.4.4—**Determine the meaning of words and phrases as they are used in a text, including those that allude to significant characters found in mythology (e.g., Herculean).
Vocabulary:	Program: An algorithm that has been coded into something that can be run by a machine Programming: The art of creating a program
Notes:	If the teacher does not have a Code & Go Mouse, a student could act as the mouse to maneuver through the maze.

Week 35: Energy Sources

Lesson overview:	Purpose: Students will create a presentation in support of the energy source that they feel is most sustainable and environmentally friendly. Lesson: ● Introduction Students watch videos and create a project in Scratch. ● Videos: Introduction to Scratch. (Use smartboard to show videos in attached slide or assign the slide to students in google classroom for them to view independently.) *Youtube has some additional instructional videos if needed. ● Demonstrate on smart board how to open the project in Scratch. ● Interactive presentation ● Main Activity ○ Build Your presentation using "Pitch your Passion" ○ Survey-interaction presentation ● Wrap Up
Lesson links/resources:	Energy Sources Lesson Slide
CS standards addressed:	Students will be able to: ● Express what they feel is the most sustainable and environment-friendly energy source through the creation of an animation Standards: ● **AP.1B.5—**Modify, remix, or incorporate portions of an existing program into one's own work to develop something new or add more advanced features. ● **AP.1B.6—**Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction **10 min** ● Main Activity **35 min** ● Wrap Up **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	Science
Other standards Addressed:	● **E.4.10.1—**Organize simple data sets to compare energy and pollution output of various traditional, nonrenewable resources (e.g., coal, crude oil, wood). ● **E.4.10.2—**Use technology or informational text to investigate, evaluate, and communicate various forms of clean energy generation

Vocabulary:	Parallelism: The process of events happening at the same time, either independently or interdependently Debugging: The process of identifying and fixing error(s) in a program when it is not functioning as expected Control structures: Sections of code that order the direction or flow of how a program functions. The control structure in this lesson focuses on loops
Notes:	This lesson was recreated using resources from CS First which is no longer available online.

Week 36: Code.org, Course E, Lesson 16—Conditionals With the Farmer

Lesson overview:	**Purpose:** This lesson introduces students to while loops and "if/else" statements. **Lesson:** ● Introduction ○ Key vocabulary ● Conditionals With the Farmer ○ Skill building ○ While loops with the farmer (video) ○ Prediction ○ Skill building ○ Repeat until blocks (video) ○ Skill building ○ "If/else" (video) ○ Challenge ○ Practice ● Wrap Up/Reflection ● Extended Learning
Lesson links/resources:	Course E Lesson 16: Conditionals With Farmer
CS standards addressed:	Students will be able to: ● Define circumstances when certain parts of a program should run and when they should not ● Determine whether a conditional is met based on criteria Standards: **1B-AP-10** - Create programs that include sequences, events, loops, and conditionals. **1B-AP-11** - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. **1B-AP-17** - Describe choices made during program development using code comments, presentations, and demonstrations.
Time needed:	**Total Time: 60 min** ● Warm Up **15 min** ● Main Activity **30 min** ● Wrap Up/Reflection **15 min**
Materials needed:	Teachers: ● Smartboard/projector with sound ● **CSF - Course E - Slides** - Slides (**Download**) ● Pause and Think Online - Video Students: ● Student devices with access to the internet
Subject integrated:	Social Studies
Other standards addressed:	● **G.4.1.1—**Compare and contrast the ten geographical regions of Mississippi in terms of soil, landforms, etc. ● **G.4.2.3—**Describe the opportunity cost of choices made within Mississippi (e.g., cotton farming vs. soybean farming, pastureland

	vs. industrial development, beaches vs. casinos, landfills vs. parks, etc.).
Vocabulary:	Condition: Something a program checks to see if it is true before allowing an action Conditionals: Statements that only run under certain conditions While Loop: A loop that continues to repeat while a condition is true
Notes:	

Week 37: Angle Measures

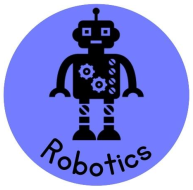
Lesson overview:	**Purpose:** Students will answer questions about angles then program a robotic mouse to run through a maze. **Lesson:** ● Introduction ○ Review terms ● Main Activity ○ Answer questions while playing Kahoot ○ Teams that answer the question correctly, get to enter a code on their robot. If it is a high-level question, it can be worth more. ● Wrap Up ○ Debugging ○ Reflection
Lesson links/resources:	● Angle Measures Lesson Plan ● Kahoot
CS standards addressed:	Students will be able to: ● Program a robotic mouse to move through a maze while answering questions Standards: ● **AP.1B.4—**Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
Time needed:	**Total Time: 60 min** ● Introduction **10 min** ● Main Activity **40 min** ● Wrap Up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound ● Robot (Some examples include: Code and Go Mouse, Botley, Dash) ● Coding maze (Should be already coded before the game) ● Grade level passage (Can be passage previously discussed) ● Access to Kahoot or Quizizz Students: ● Student devices with access to the internet
Subject integrated:	Math
Other standards addressed:	**4.MD.7—**Recognize angle measure as additive. When an angle is decomposed into non overlapping parts, the angle measure of the whole is the sum of the angle measures of the parts. Solve addition and subtraction problems to find unknown angles on a diagram in real world and mathematical problems, e.g., by using an equation with a symbol for the unknown angle measure
Vocabulary:	Program: An algorithm that has been coded into something that can be run by a machine

	Programming: The art of creating a program
Notes:	

Week 38: Text Structures (Coding)

Lesson overview:	**Purpose:** Today, students will create animated characters that will discuss text structures. The students will use Scratch to discuss text structures. **Lesson:** ● Introduction ○ The teacher will review using Scratch with students. ● Main Activity ○ The students will create different animations where the characters are discussing the different text structures. ● Wrap Up ○ The students can share their animations with the group ○ Reflection
Lesson links/resources:	Scratch
CS standards addressed:	Students will be able to: ● Define circumstances when certain parts of a program should run and when they should not ● Determine whether a conditional is met based on criteria Standards: ● **AP.1B.4—**Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process
Time needed:	**Total Time: 60 min** ● Introduction **10 min** ● Main Activity **40 min** ● Wrap Up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound Students: ● Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	**RI.4.5—**Determine the overall structure (chronology, comparison, cause/effect, etc.) of events, ideas, or information in a text or pair of text.
Vocabulary:	
Notes:	

Week 39: Text Structure (Robotic)

Lesson overview:  	**Purpose:** Students will answer questions about Text Structure then program a robotic mouse to run through a maze. The teacher will need to pull questions on text features for this activity. **Lesson:** ● Introduction ○ Review text structures ● Main Activity ○ Answer questions while playing Kahoot ○ Teams that answer the question correctly get to enter a code on their robot. If it is a high-level question, it can be worth more. ● Wrap Up ○ Debugging ○ Reflection
Lesson links/resources:	● Text Structure (Robotic) Lesson Plan ● Kahoot on Text Structure
CS standards addressed:	Students will be able to: ● Program a robotic mouse to work through a maze Standards ● **AP.1B.1—**Compare and refine multiple algorithms for the same task and determine which is the most appropriate. ● **AP.1B.4—**Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
Time needed:	**Total Time: 60 min:** ● Introduction **10 min** ● Main Activity **40 min** ● Wrap Up **10 min**
Materials needed:	Teachers: ● Smartboard/projector with sound ● Robot (Some examples include: Code and Go Mouse, Botley, Dash) ● Maze mat ● Coding maze (Should be already coded before the game) ● Grade level passage: (Can be passage previously discussed) ● Access to Kahoot or Quizizz Students: ● Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	**RI.4.5—**Describe the overall structure (e.g., chronology, comparison, cause/effect, problem/solution) of events, ideas, concepts, or information in a text or part of a text.

Vocabulary:	Program: An algorithm that has been coded into something that can be run by a machine Programming: The art of creating a program
Notes:	

Week 40: Code.org, Course E, Lesson 17– Designing for Accessibility

Lesson overview: Coding	**Purpose:** Through learning about accessibility, students recognize the impacts of computing beyond their own lives. Accessibility might not seem like a relevant CS topic, but creating technology that is accessible for underserved users helps make tech better for everyone else as well. **Lesson:** ● Introduction – Discuss the question, What are apps? ● Designing for Accessibility ● Scenarios – Students will work in groups to explain what the text says explicitly and then draw inferences from the text to design an app to address the accessibility needs of users. ● Designing for Accessibility Scenarios ● App Redesign Activity ● Reflection/Share
Lesson links/resources:	Course E Lesson 17: Designing for Accessibility
CS standards addressed:	Students will be able to: ● Describe the impact of mobile apps on the modern world. ● Explain why accessibility is an important part of designing an app for users. ● Improve upon an existing app design by addressing the accessibility needs of users. Standards: ● **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. ● **1B-AP-13** - Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences. ● **1B-CS-03** - Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. ● **1B-IC-18** - Discuss computing technologies that have changed the world and express how those technologies influence, and are influenced by, cultural practices. ● **1B-IC-19** - Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users. ● **1B-IC-20** - Seek diverse perspectives for the purpose of improving computational artifacts.

Time needed:	**Total Time:** **60 min** ● Warm Up/Introduction **5 min** ● Main Activity **35 min** ● Wrap Up/Reflection/Share **5 min** ● Keyboarding Practice **15 min**
Materials needed:	Teachers: • **Smartboard/projector with soundFor the teachers** • **CSF - Course E - Slides - Slides (Download)** • **Types of Disabilities – Resource** Students: ● Student devices with access to the internet
Subject integrated:	ELA
Other standards addressed:	RL.4.1 Refer to details and examples in a text when explaining what the text says explicitly and when drawing inferences from the text.
Vocabulary:	
Notes:	

Week 41: Code.org, Course E, Lesson 18—Digital Sharing	
Lesson overview: For the students Unplugged Digital Citizenship	**Purpose:** Students will soon be creating projects to share and most of these projects will contain either code or imagery that students did not create themselves. This lesson is here to show students the proper way to handle the use of content that is not their own. **Lesson:** ● Introduction ○ What are examples of projects or artwork that people share online? ● Write a Character Sketch – Use slides provided ● Vocabulary (Copyright) ● Digital Sharing - video ● Reflection ● Keyboarding Practice ○ If time remains, have students use resources from above to practice keyboarding.
Lesson links/resources:	Course E Lesson 18: Digital Sharing
CS standards addressed:	Student will be able to: ● Interpret ethical sharing of copyrighted material vs. sharing that is not ethical. ● Understand their own rights regarding materials that they have created Standards: ● **1B-AP-14** - Observe intellectual property rights and give appropriate attribution when creating or remixing programs. ● **1B-IC-21** - Use public domain or creative commons media and refrain from copying or using material created by others without permission.
Time needed:	**Total Time: 60 min** ● Warm Up/Introduction 1**5 min** ● Main Activity **30 min** ● Wrap Up **10 min** ● Keyboarding Practice **5 min**
Materials needed:	Teachers: ● **CSF - Course E - Slides** - Slides (**Download**) ● **Digital Sharing Lesson Plan** – Resource ● Smartboard/projector with sound

	Students: - **Digital Sharing Ethics (Video)** - Video
Subject integrated:	Writing
Other standards addressed:	**W.4.6—**With some guidance and support from adults, use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills to type a minimum of one page in a single sitting.
Vocabulary:	**Copyright** - the exclusive legal right to print, publish, perform, film, or record literary, artistic, or musical material, and to authorize others to do the same
Notes:	