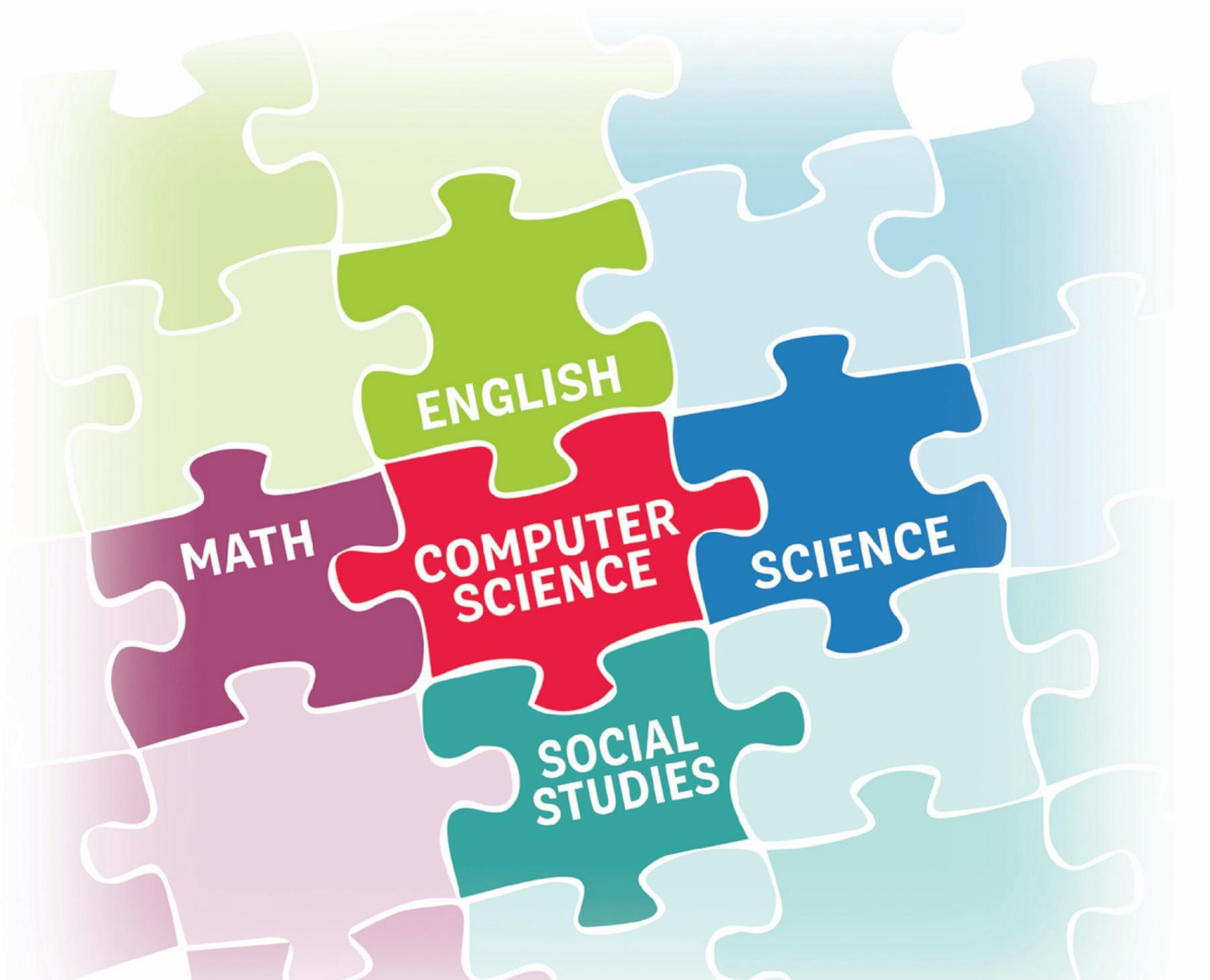


2022 Elementary Integration Guide

THIRD GRADE

c spire

MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

Acknowledgements

The following people assisted in the development of this integration guide:

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:

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. 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 ◦ 3rd Grade: Course D • Common Sense Digital Media • Scratch
CS4MS website materials	• 2018 Mississippi Computer Science Standards • CS4MS Website
Keyboard practice	• Astro Bubbles Keyboard Practice • Nitrotype • Typing.com
Teacher/student accounts	• Code.org • Common Sense Digital Media • Scratch
Help with this guide	• Contact Mississippi State University's Center for Cyber Education: www.tinyurl.com/ccehelpdesk

Contents by Integrated Subjects

ELA

Math

Science

- Week 36: E.3.9—Earth’s systems
- Week 37: 3.LS1.1 (Next Gen) Life cycles

Social Studies

- Week 3: CI.3.1—Rights/responsibilities of local government
- Week 5: CI.3.2—Community/local government
- Week 44: H.3.2—Branches of government

Contents by Topics

Coding

Digital Citizenship

Digital Literacy

Keyboarding

Robotics

Unplugged

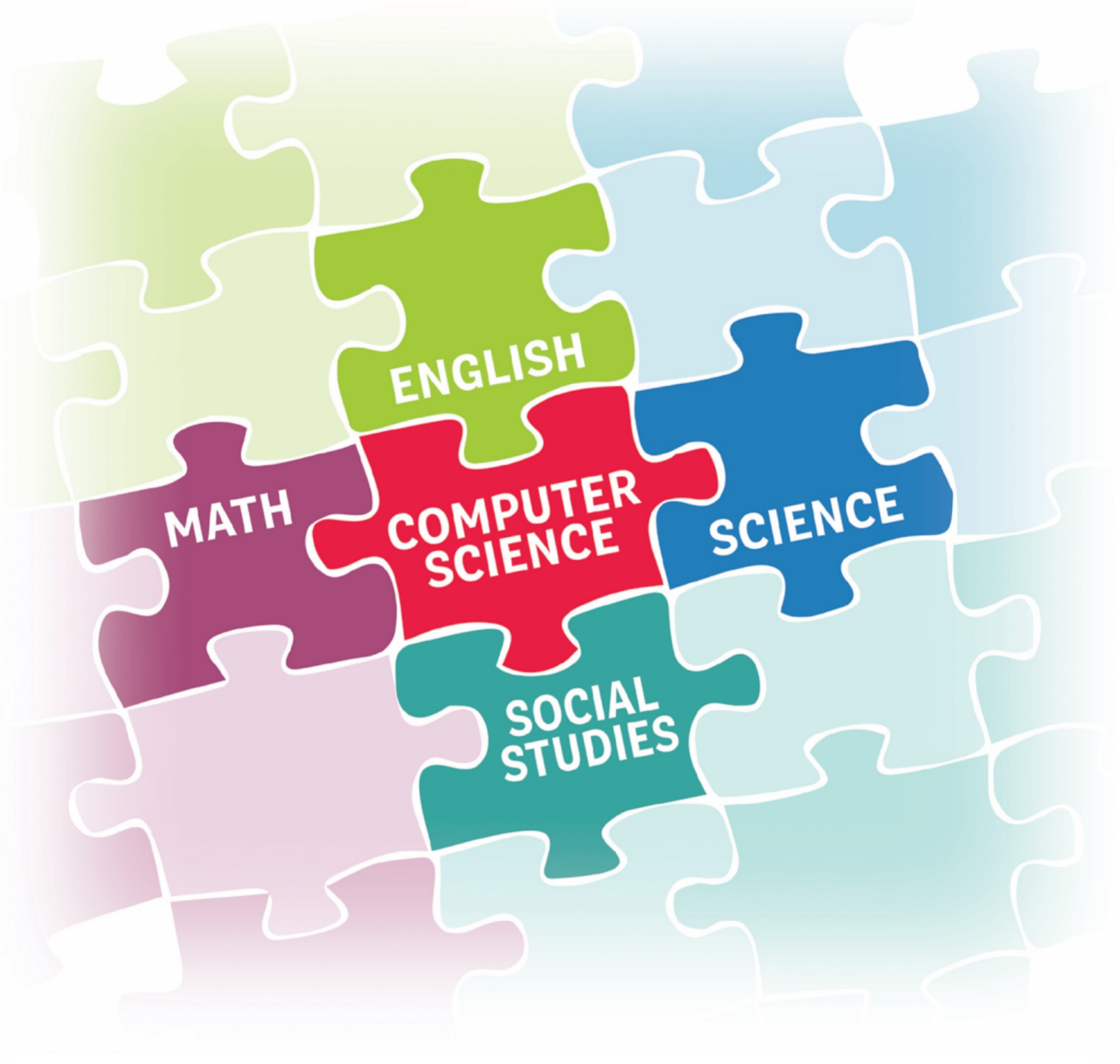
Calendar/ Pacing Per Week:

⟶ 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	Identifying Parts of a Computer	Keyboarding	CS.1A.2	SL.3.1	ELA
2	Keyboarding Practice	Keyboarding	CS.1A.2	RI.3.7	ELA
3	Your Rings of Responsibility → Account creation needed	Digital Citizenship	1C.1B.4	CI.3.1	Social Studies
4	This is Me	Digital Citizenship	N1.1B.1	SL.3.1	ELA
5	Our Digital Citizenship Pledge	Digital Citizenship	N1.1B.1	CI.3.2	Social Studies
6	Password Power-Up → Account creation needed	Digital Literacy Unplugged	N1.1B.2	SL.3.1	ELA
7	Graph Paper Programming	Coding Unplugged	AP.1B.4	SL.3.1	ELA
8	Introduction to Online Puzzles	Coding	AP.1B.4	W.3.3c	ELA
9	Relay Programming	Coding Unplugged	AP.1B.4 AP.1B.5 AP.1B.8	3.NBT.1	Math
10	Debugging With Laurel	Coding	AP.1B.4	SL.3.1	ELA
11	Events in Bounce	Coding	AP.1B.5 CS.1B.1 CS.1B.1	SL.3.1 3.MD.4 CR.3.3	ELA Math Social Studies
12	Game Day Commentary	Coding	AP.1B.5 CS.1B.1 CS.1B.2	RF.3.4 SL.3.4 3.MD.3	ELA Math
13	Build a Star Wars Game	Coding	AP.1B.4	RL.3.3	ELA
14	Dance Party	Coding	AP.1B.2/AP.1B 3	L3.4	Science
15	Digital Citizenship—The Power of Words	Digital Citizenship	NI.1A.2	RL.3.4	ELA
16	Loops in Ice Age	Coding	AP.1B.2 AP.1B.3 AP.1B.4 AP.1B.7	3.OA.9	Math
17	Drawing Shapes with Loops	Coding	AP.1B.2 AP.1B.3 AP.1B.4 AP.1B.8	3.OA.9	Math

18	Nested Loops in Maze	Coding	AP.1B.4 AP.1B.5 AP.1B.8	3.OA.9	Math
19	Conditionals With Cards	Coding Unplugged	AP.1B.4	SL.3.3 3.OA.7	ELA Math
20	Play the Deck With Conditionals	Coding Unplugged	AP.1B.4	SL.3.3 3.OA.7	Math ELA
21	Looking Ahead With Minecraft	Coding	AP.1B.3	RI.3.8	ELA
22	If/Else With Bee	Coding	AP.1B.4	W.3.3	ELA
23	Loops in Farmer	Coding	AP.1B.4	3.OA.9	Math
24	Until Loops in Maze	Coding	AP.1B.4	3.OA.9	Math
25	Dialogue → Account creation needed	Coding	B.AP.10 B.AP.11 B.AP.12	RL.3.2	ELA
26	Check It Out	Coding	B.AP.10 B.AP.11 B.AP.12	RL.3.3	ELA
27	Setting	Coding	B.AP.10 B.AP.11 B.AP.12	RL.3.3 RL.3.5	ELA
28	Lesson 1 of 2: Premise	Coding	B.AP.10 B.AP.11 B.AP.12	RL.3.3 RL.3.5	ELA
29	Lesson 2 of 2: Premise	Coding	B.AP.10 B.AP.11 B.AP.12	RL.3.3 RI.3.5	ELA
30	Lesson 1 of 2: Characterization	Coding	B.AP.10 B.AP.11 B.AP.12	RL.3.3	ELA
31	Lesson 2 of 2: Characterization	Coding	B.AP.10 B.AP.11 B.AP.12	RL.3.3	ELA
32	Interactive Storytelling	Coding	B.AP.10 B.AP.11 B.AP.12	RL.3.3	ELA
33	Personal Narrative	Coding	B.AP.10 B.AP.11 B.AP.12	W.3.3	ELA
34	Story of Innovation	Coding	B.AP.10 B.AP.11 B.AP.12	W.3.2	ELA
35	Ecosystems in Sprite Lab	Coding	B.AP.10 B.AP.11 B.AP.12	E.3.9	Science

36	Life Cycles in Sprite Lab	Coding	B.AP.10 B.AP.11 B.AP.12	3-LS.1.1	Science
37	Character Study in Sprite Lab	Coding	B.AP.10 B.AP.11 B.AP.12	RL.3.3 RL.3.5	ELA
38	Story Morals in Sprite Lab	Coding	B.AP.10 B.AP.11 B.AP.12	RL.3.3 RL.3.5	ELA
39	Find the Area/Robot Game	Coding Robot Unplugged	AP.1B.8 AP.1B.9 AP.1B.10	3.MD.7	Math
40	Character Traits Robot Game	Coding Unplugged	B.AP.10 B.AP.11 B.AP.12	RL.3.3	ELA
41	The Role of the Three Branches of Government	Coding Robot Unplugged	AP.1B.8 AP.1B.9 AP.1B.10	H.3.2	Social Studies



Lessons and Activities

THIRD GRADE

Week 1: Identifying Parts of a Computer

Lesson overview: Digital Literacy	**Purpose:** In this lesson, students will learn the basic parts of computers, including the computer, monitor, CPU, keyboard, mouse, printer, and modem/router. Students will be introduced to technology they will use throughout school. **Lesson:** ● Students will use the Parts of a Computer Blank PDF worksheet to write the definition of each part. The teacher will use the Parts of a Computer Key PDF to provide the definitions. ● Students will play a game called "Who Am I?" in which the teacher gives clues about a particular computer part, and students identify the correct term based on the clues. ○ Parts of a computer—Who am I? Have the students guess who you are by reading the following statements: ■ I have letters, numbers, and words on me. Users press me, and I appear in front of you. Who am I? ■ I am at eye level when you are sitting correctly. I have a frame around me. My image is always changing depending on what you are doing. Who am I? ■ Any work you save or programs you have are stored in me so you can always access them. Who am I? ● The teacher will access the Find the Technology game if time allows. This can be played as a group or individually. Some items in the game are not listed on the students' worksheets.
Lesson links/resources:	● Parts of a Computer Blank PDF ● Parts of a Computer Key PDF ● Find the Technology game
CS objectives addressed:	Students will be able to: ● Identify parts of the computer ● Identify various types of technology Standards: ● **CS.1A.2**—Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).
Time needed:	**Total Time: 60 min** ● Identifying the parts of a computer and completing the Parts of a Computer Blank PDF **30 min** ● "Who Am I?"—Repeat allowing students an opportunity to create "Who Am I?" statements **10 min** ● Online Find the Technology game **20 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Parts of a Computer Key PDF Students: ● Parts of a Computer Blank PDF ● Computer/tablet with internet access

	Subject integrated:	ELA
	Other standards addressed:	**SL.3.1**—Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 3 topics and texts, building on others' ideas and expressing their own clearly.
	Vocabulary:	Computer: A device for working with information Central Processing Unit (CPU): The brains of the computer Monitor: Displays information on a screen Desktop Tower (Case): Contains the brains of the computer that processes information, including the CPU and other parts Keyboard: A device for typing letters, numbers, and symbols into the computer Mouse: A hand-held device for moving a cursor around the screen Laptop: A small, portable personal computer Tablet: A wireless, portable personal computer with a touchscreen interface
	Notes:	Could play the "Who am I?" game as a matching game.

Week 2: Keyboarding Practice

Lesson overview: Keyboarding Digital Literacy	**Purpose:** **Introduction to Keyboards** The computer keyboard activities in this lesson are designed to help beginning computer users learn keyboard skills through hand-eye coordination by finding letters/numbers/symbols and using a finger to press the key. Students will become familiar with finding letters/numbers/symbols on the QWERTY keyboard layout. **Lesson** **Keyboard Coloring Sheet:** ● Introduction ○ Show students an image of a keyboard or model the keyboard on their devices. Show them how to properly place their hands and the QWERTY keyboard layout. ○ Show the Keyboard Rap for additional enrichment. ● Students will label the letter for each key on the Keyboard L-R Coloring Sheet. ● Students will identify the left and right sides of the keyboard and learn which hands to use. ○ Students will color the left side of the keyboard to coordinate with the left hand and color the right side of the keyboard to coordinate with the right hand. ○ The teacher will call out various letters/numbers to help students practice finding letters/numbers on a keyboard. ● Students will play Typing Games to begin familiarizing themselves with letter placement. ○ Provide an email or online announcement, including the hyperlink, for the typing game and explain expectations. Teach students how to read a text, email, or announcement and use a hyperlink (Please see ELA standard **RI.3.7**).
Lesson links/resources:	● Keyboard L-R Coloring Sheet ● Typing Games ● Keyboard Rap
CS standards addressed:	Students will be able to: ● Identify and label parts of keyboard ● Develop their keyboarding skills Standards: ● **CS.1A.2**—Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).
Time needed:	**Total Time: 60 min** ● Introduction to the keyboard and teacher modeling **10 min** ● Keyboard L-R Coloring Sheet/Guided Practice—The teacher will have students identify keys on the keyboard, show proper hand placement, and practice typing. **20 min** ● Typing Games **30 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Keyboard Rap

	Students: ● Computer/tablet with internet access ● Markers, crayons, color pencils, etc. ● Keyboard L-R Coloring Sheet ● Typing Games
Subject integrated:	ELA
Other standards addressed:	**RI.3.5**—Use text features and search tools (e.g., key words, sidebars, hyperlinks) to locate information relevant to a given topic efficiently.
Vocabulary:	Keyboard: A device for typing letters, numbers, and symbols into the computer
Notes:	Could play a game with printed out keyboards where teacher callsl out a letter or number and students tell which hand and finger they used to type the key.

Week 3: Your Rings of Responsibility

Lesson overview:	**Purpose:** How do digital citizens take responsibility for themselves, their communities, and their world? **Lesson:** ● Students will examine in-person and online responsibilities, describe the Rings of Responsibility as a way to think about how our behavior affects ourselves and others, and identify examples of online responsibilities to others. ● Students can practice keyboarding using the links in the previous lesson if time allows.
Lesson links/resources:	Rings of Responsibility lesson
CS standards addressed:	Students will be able to: ● Develop their keyboarding skills ● Develop an understanding of digital citizenship by being responsible online Standards: ● **IC.1B.4**—Use public domain or creative commons media and refrain from copying or using material created by others without permission.
Time needed:	**Time Needed: 60 min** ● Rings of Responsibility lesson **45 min** ● Keyboard practice **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Rings of Responsibility lesson ● Common Sense teacher account Students: ● Computer/tablet with internet access ● Common Sense teacher account
Subject integrated:	Social Studies
Other standards addressed:	**CI.3.1**—Explain how an individual exercises rights and responsibilities within the community and local government.
Vocabulary:	Digital citizen: Someone who uses technology responsibly to learn, create, and participate
Notes:	→Teachers will need to create free teacher and/or student accounts (when applicable) at commonsense.org.

Week 4: This is Me

Lesson overview:	**Purpose:** How does what I post online affect my identity? **Lesson:** - Students will consider how posting selfies or other images will lead others to make assumptions about them, reflect on the most important parts of their unique identity, and identify ways they can post online to best reflect who they are. - Students can practice keyboarding using the links in the previous lesson if time allows.
Lesson links/resources:	This is Me lesson
CS objectives addressed:	Students will be able to: - Enhance their understanding of digital citizenship to be responsible online. - Develop their keyboarding skills - Use the mouse and keyboard Standards: - **NI.1B.1**—Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the internet, and reassembled at the destination.
Time needed:	**Total Time: 60 minutes** - This is Me lesson**45 min** - Keyboard practice **15 min**
Materials needed:	Teacher: - Computer - Projector/smartboard with sound - This is Me lesson - Common Sense teacher account Students: - Computer/tablet with internet access - Common Sense teacher account
Subject integrated:	ELA
Other standards addressed:	**SL.3.1**—Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 3 topics and texts, building on others' ideas and expressing their own clearly.
Vocabulary:	Selfie: A picture you take of yourself, usually with a phone
Notes:	

Week 5: Our Digital Citizenship Pledge

Lesson overview:	**Purpose:** What makes a strong online community? **Lesson:** ● Students will define what a community is, both in person and online, explain how having norms helps people in a community achieve their goals, and create and pledge to adhere to shared norms for being in an online community. ● Students can practice keyboarding using the links in the previous lesson if time allows.
Lesson links/resources:	Our Digital Citizenship Pledge lesson
CS objectives addressed:	Students will be able to: ● Develop their digital citizenship skills by understanding the importance of online etiquette ● Develop their keyboarding skills Standards: ● **NI.1B.1**—Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the internet, and reassembled at the destination.
Time needed:	**Total Time: 60 minutes** ● Our Digital Citizenship Pledge lesson **45 min** ● Keyboard practice **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Our Digital Citizenship Pledge lesson ● Common Sense teacher account Students: ● Computer/tablet with internet access ● Common Sense teacher account
Subject integrated:	Social Studies
Other standards addressed:	**CI.3.2**—Demonstrate knowledge of community and local government.
Vocabulary:	Digital citizen: Someone who uses technology responsibly to learn, create, and participate
Notes:	

Week 6: Code.org, Course D,Lesson 1—Password Power-Up

Lesson overview: Digital Literacy Unplugged	**Purpose:** Common Sense Education created this lesson to teach students how strong passwords can help protect their privacy. **Lesson:** ● Warm Up: ○ Pssst ... What's a Password? ○ Key Vocabulary ● Evaluate: ○ Uh-Oh! If ... Then ... ● Create: ○ Power Up Your Password ● Wrap Up: ○ Password Tips Notes ● Extended Learning
Lesson links/resources:	Lesson 1: Password Power-Up
CS objectives addressed:	Students will be able to: ● Define the term "password" and describe a password's purpose ● Practice creating a memorable and strong password Standards: ● **NI.1B.2—**Discuss real-world cybersecurity problems and how personal information can be protected.
Time needed:	**Total Time: 60 min** ● Warm Up **10 min** ● Evaluate **20 min** ● Create **10 min** ● Wrap Up **5 min** ● Extended Learning **~15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Password Power-Up: If...Then...Scenarios—Answer key ● Password Power-Up: Lesson Quiz—Answer key ● Password Power-Up: Lesson Slides— Slides ● Password Power-Up: Power-Up Your Password—Answer key ● Code.org account Students: ● Computer/tablet with internet access ● Password Power-Up: If...Then...Scenarios—Handout ● Password Power-Up: Lesson Quiz—Resource ● Password Power-Up: Power-Up Your Password—Handout ● Code.org account
Subject integrated:	ELA
Other standards addressed:	**SL.3.1—**Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 3 topics and texts, building on others' ideas and expressing their own clearly.
Vocabulary:	Password: A secret string of letters, symbols, and numbers that you can use

	to restrict who can access something digital Phrase: A group of words that go together and are easy to remember Symbol: A character other than a number or letter, such as #, !, or @ Username: A name you create to sign into a website, app, or game
Notes:	Build in with classroom routines and procedures. →Teachers will need to create free teacher and/or student accounts (when applicable) at code.org.

Week 7: Code.org, Course D, Lesson 2—Graph Paper Programming

Lesson overview: Coding Unplugged	**Purpose:** In this context-setting lesson, students use symbols to instruct each other to color squares on graph paper. By "programming" one another to draw pictures, students get an opportunity to experience some of the core concepts of programming in a fun, accessible way. This activity aims to build critical thinking skills and excitement for the course while introducing some of the fundamental programming concepts that students will use 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:** ● Warm Up ○ Introduction ● Main Activity ○ Graph Paper Programming ● Wrap Up ○ Reflection ● Extended Learning
Lesson links/resources:	Lesson 2: Graph Paper Programming
CS objectives 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: 65-70 min** ● Warm Up **10 min** ● Main Activity **30 min** ● Wrap Up **15 min** ● Extended Learning **10-15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Graph Paper Programming—Lesson in action video ● Graph Paper Programming—Worksheet answer key ● Graph Paper Programming—Assessment answer key ● Code.org account Students: ● Computer/tablet with internet access ● Graph Paper Programming—Activity worksheet ● Graph Paper Programming—Unplugged video (Download) ● Graph Paper Programming—Assessment ● Code.org account

Subject integrated:	ELA
Other standards addressed:	**SL.3.1—**Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 3 topics and texts, building on others' ideas and expressing their own clearly.
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:	● Could use Code & Go Robot Mouse with this lesson ● Team-building activity ● Could use tile floor and index cards to make different algorithms to move around the room. (addition facts) (multiplication facts)

Week 8: Code.org, Course D, Lesson 3—Introduction to Online Puzzles

Lesson overview:	**Purpose:** ● We recognize that 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. To create an equal playing (and learning) field, we have 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. In this skill-building lesson, students will practice their sequencing and debugging skills in maze puzzles. **Lesson:** ● Warm Up ○ Introduction ● Bridging Activities—Programming ○ Preview of Online Puzzles as a Class (may project on Promethean board to see how drag and drop of blocks works.) ● Main Activity ○ Online Puzzles ● Wrap Up ○ Reflection
Lesson links/resources:	Lesson 3: Introduction to Online Puzzles
CS objectives 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 **10 min** ● Bridging Activities **10 min** ● Main Activity **30 min** ● Wrap Up **10 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Pair Programming—Student video ● Code.org account
Subject integrated:	ELA
Other standards	**W.3.3.C**—Use temporal words and phrases to signal event order.

addressed:	
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 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 9: Code.org, Course D, Lesson 4—Relay Programming

Lesson overview: Coding Unplugged	**Purpose:** Teamwork is essential in computer science. Teams write and debug code with each other instead of working as individuals. In this lesson, students will learn to work together while being as efficient as possible. This activity also provides a sense of urgency that will teach students to balance their time carefully and avoid mistakes without falling too far behind. This experience can be stressful (which is expected!). Make sure you provide students with the tools to deal with potential frustration. Teams write and debug code with each other instead of working as individuals. In this lesson, students will learn to work together while being as efficient as possible. **Extended Activity:** Code & Go Robot Mouse (or other codable robot) ● Label a grid with three-digit numbers. Ask students to locate the number on the grid that rounds to a particular number. Students will program the mouse to get to the correct answer. **Lesson:** ● Warm Up ○ Where did I go wrong? ● Main Activity ○ Relay Programming ● Wrap Up ○ Reflection ● Extended Learning
Lesson links/resources:	Lesson 4: Relay Programming
CS objectives addressed:	Students will be able to: ● Define ideas using code and symbols ● Identify signs of frustration ● Verify work done by teammates Standards: ● **AP.1B.4**—Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. ● **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.8**—Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
Time needed:	**Total Time: 60-70 min** ● Warm Up **15 min** ● Main Activity **20 min** ● Wrap Up **15 min** ● Extended Activity **~ 20 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Relay Programming—Worksheet answer key ● Relay Programming—Teacher debugging image

	● Relay Programming—Teacher video ● Code.org account Students: ● Computer/tablet with internet access ● Relay Programming—Worksheet ● Relay Programming—Unplugged video (Download) ● Relay Programming—Activity packet ● Code.org account
Subject integrated:	Math
Other standards addressed:	**3.NBT.1**—Use place value understanding to round whole numbers to the nearest 10 or 100.
Vocabulary:	Algorithm: A list of steps to finish a task Bug: Part of a program that does not work correctly Debugging: Finding and fixing problems in an algorithm or program Program: An algorithm that has been coded into something that can be run by a machine
Notes:	

Week 10: Code.org, Course D, Lesson 5—Debugging With Laurel	
Lesson overview:	**Purpose:** The purpose of this lesson is to teach students that failure is normal when learning a new skill. Students will be given pre-written programs that do not work. They will be asked to fix these programs. This process, called debugging, teaches students essential problem-solving and critical thinking skills. These skills transfer over as students proceed to harder and harder programming projects. In this skill-building lesson, students will practice debugging in the "collector" environment. Students will get to practice reading and editing code to fix puzzles with simple algorithms, loops, and nested loops. Be sure to choose just one of the bridging activities. **Lesson:** ● Warm Up ● Bridging Activities (Choose One) ○ Unplugged Activity with Paper Blocks ○ Preview of Online Puzzles as a Class ● Main Activity ○ Debugging with Laurel ● Wrap Up ○ Reflection
Lesson links/resources:	Lesson 5: Debugging with Laurel
CS objectives addressed:	Students will be able to: ● Describe and implement a plan to debug a program ● Identify a bug and the problems it causes in a program ● Read and comprehend given code Standards: ● **AP.1B.4**—Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
Time needed:	**Total Time: 60-75 min** ● Warm Up **15 min** ● Bridging Activities **15 min** ● Main Activity **30 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Slides ● Activity handouts ● Code.org account Students: ● Computer/tablet with internet access ● Unplugged Blocks (Courses C-F)—Manipulatives ● Relay Programming—Activity packet ● Code.org account
Subject integrated:	ELA

	Other standards addressed:	**SL.3.1**—Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 3 topics and texts, building on others' ideas and expressing their own clearly.
	Vocabulary:	Bug: Part of a program that does not work correctly Debugging: Finding and fixing problems in an algorithm or program
	Notes:	Could have a robot day during this time and give a code for the Code and Go Mouse or Botley with an intentional bug for students to figure out.

Week 11: Code.org, Course D, Lesson 6—Events in Bounce

Lesson overview:	**Purpose:** Events are very common in computer programs, especially in video games. In this lesson, students will develop their understanding of events by making a sports-based game. Students will learn to make their paddle move according to arrow keys and make noises when objects collide. At the very end, they will get to customize their game to make it more unique! In this context-setting/skill-building lesson, students will learn what events are and how programmers use them in video games. Students will build a game that they can customize with different speeds and sounds. Lesson: ● Warm Up ○ Introduction ● Main Activity ○ Events in Bounce ● Wrap Up ○ Reflection ○ How Events Happen ● Extended Learning ○ Take Me Out to the Ballgame (found at the end of Lesson 6: Events in Bounce)
Lesson links/resources:	Lesson 6: Events in Bounce
CS objectives addressed:	Students will be able to: ● Create an interactive game using sequence and event handlers ● Identify actions that correlate to input events ● Share a creative artifact with other students 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 ● **CS.1B.1**—Describe how internal and external parts of computing devices function to form a system. ● **CS.1B.2**—Model how computer hardware and software work together as a system to accomplish tasks.
Time needed:	**Total Time: 60-80 min** ● Warm Up **10 min** ● Main Activity **30 min** ● Wrap Up **10 min** ● Extended Learning **~30 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Events in Bounce How do they Happen? ● Open-ended Programming Levels—Answer key ● Code.org account Students: ● Computer/tablet with internet access ● Pause and Think Online—Video ● Code.org account

	Subject integrated:	ELA Math Social Studies
	Other standards addressed:	ELA ● **SL.3.1**—Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 3 topics and texts, building on others' ideas and expressing their own clearly. Math ● **3.MD.3**—Draw a scaled picture graph and a scaled bar graph to represent a data set with several categories. ● **3.MD.4**—Generate measurement data by measuring lengths using rulers marked with halves and fourths of an inch. Social Studies ● **CR.3.3**—Examine Declaration of Independence, Constitution, and Bill of Rights in order to recognize basic principles of democracy and civil liberties.
	Vocabulary:	Event: An action that causes something to happen
	Notes:	If students are struggling with coding, you could do some coding as whole group.

	Week 12: Code.org, Course D, Lesson 6—Game Day Commentary	
	Lesson overview:	**Purpose:** Game Day Commentary is an activity aligned to Common Core ELA, Common Core Math, and Next Generation Science Standards, written by our teacher community. In this activity, students will read aloud instructions as they learn to code with events in Bounce to make a game. Once the game is complete, students will provide play-by-play commentary as their classmates predict which way the ball will bounce in order to score. Students will also complete a Frayer model about bar graphs prior to creating a scaled bar graph using the data collected during their game. **Lesson:** ○ Pair program ○ Student swap ○ Complete Frayer model ○ Create bar graph ○ Closure
	Lesson links/resources:	Game Day Commentary Lesson
	CS objectives addressed:	Students will be able to: ● Create an interactive game using sequence and event handlers ● Identify actions that correlate to input events ● Share a creative artifact with other students 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. ● **CS.1B.1**—Describe how internal and external parts of computing devices function to form a system. ● **CS.1B.2**—Model how computer hardware and software work together as a system to accomplish tasks.
	Time needed:	**Total Time: 45-60 min** ● Pair program **25-30 min** ● Student swap **15 min** ● Complete Frayer model **5 min** ● Create bar graph **5 min** ● Closure **5 min**
	Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Game Day Commentary: Frayer Model ● Colored pencils/crayons and paper (optional) ● Code.org account
	Subject integrated:	ELA Math

	Other standards addressed:	ELA ● **RF.3.4**—Read with sufficient accuracy and fluency to support comprehension. ● **SI.3.4**—Report on a topic or text, tell a story, or recount an experience with appropriate facts and relevant, descriptive details, speaking clearly at an understandable pace. Math ● **3.MD.3**—Draw a scaled picture graph and a scaled bar graph to represent a data set with several categories.
	Vocabulary:	Event: An action that causes something to happen
	Notes:	

Week 13: Code.org, Course D, Lesson 7—Build a Star Wars Game

Lesson overview:	**Purpose:** In this lesson, students will learn about events using popular characters from Star Wars. These puzzles blur the lines between "learning" and "fun." Also, students will learn to recognize regular programming practices in games so that when they play games at home, they can see common computer science principles being used. **Lesson:** ● In this skill-building lesson, students will practice using events to build a game that they can share. ● Warm Up ○ Introduction ● Main Activity ○ Build a Star Wars Game ● Wrap Up ○ Reflection
Lesson links/resources:	Lesson 7: Build a Star Wars Game
CS objectives addressed:	Students will be able to: ● Create an animated, interactive game using sequence and events ● Identify actions that correlate to input events 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 **15 min** ● Main Activity **30 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Open-ended Programming Levels—Answer Key ● Code.org account Students: ● Computer/tablet with internet access ● Pause and Think Online—Video ● Code.org account
Subject integrated:	ELA
Other standards addressed:	**RL.3.3**—Describe characters in a story (e.g., their traits, motivations, or feelings) and explain how their actions contribute to the sequence of events.
Vocabulary:	Event: An action that causes something to happen

Notes:	This lesson can be broken down into two sessions.

Week 14: Code.org, Course D, Lesson 8—Dance Party	
Lesson overview:	**Purpose:** This lesson introduces the core CS concepts of coding and event programming (using blocks). In this skill-building lesson, students will program an interactive dance party. **Lesson:** • Getting Started • Setting the Stage ○ Activity ○ Code Your Own Dance Party ○ Level by Level Support • Wrap Up ○ Reflection ○ Assessment • Cross-Curricular ○ Survival of the Dancers
Lesson links/resources:	Lesson 8: Dance Party
CS objectives addressed:	Students will be able to: • Create dance animations with code • Develop programs that respond to timed events • Develop programs that respond to user input 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: 87 min** • Setting the Stage **5 min** • Activity **30** • Warm Up **7 min** • Cross-Curricular **45 min**
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Open-ended Programming Levels—Answer Key • Spotify playlist (all ages) • Code.org account Students: • Computer/tablet with internet access • Dance Party Project Guide—Worksheet • Dance Party: Unplugged—Resource • Code.org account
Subject integrated:	Science
Other standards	**L.3.4**—Analyze and interpret data to explain how variations in

addressed:	characteristics among organisms of the same species may provide advantages in surviving, finding mates, and reproducing (e.g., plants with larger thorns being less likely to be eaten by predators or animals with better camouflage colorations being more likely to survive and bear offspring).
Vocabulary:	Program: An algorithm that has been coded into something that can be run by a machine Code: To write an algorithm or to write instructions for a computer.
Notes:	This lesson can be divided into different sessions.

Week 15: Digital Citizenship—The Power of Words

Lesson overview:	**Purpose:** In this lesson, students will learn it is important to think about the words we use, identify ways to respond to mean words online, and decide what kinds of statements are appropriate to say online. Lesson: ● Warm Up ○ Introduce the lesson using the Same Word, Different Meaning activity ● Main Activity ○ Watch the Power of Words video ○ Complete the Words Can Hurt activity ● Wrap Up ○ Cross the Line game ○ Lesson quiz ● Online Keyboarding Practice
Lesson links/resources:	The Power of Words lesson
CS objectives addressed:	Students will be able to: ● Understand that it's important to think about the words we use because everyone interprets things differently ● Identify ways to respond to mean words online using S-T-O-P ● Decide what kinds of statements are OK to say online and which are not Standards: ● **N1.A.2**—Students should understand that computers connect them to people, places, and things around the world.
Time needed:	**Total Time: 45-60 min** ● Warm Up **10 min** ● Video and Discussion **15-20 min** ● Handout **10 min** ● Wrap Up **10 min** ● Lesson Quiz **5 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● A long strip of paper or rope ● Common Sense teacher account Students: ● Computer/tablet with internet access ● Words Can Hurt handout ● Lesson quiz ● Lesson slides ● Common Sense student account
Subject integrated:	ELA
Other standards addressed:	**RL.3.4**—Determine the meaning of words and phrases as they are used in a text, distinguishing literal from nonliteral language. I can determine the

		meaning of words and phrases that are used in a text.
	Vocabulary:	Digital citizenship: Someone who uses technology responsibly to learn, create, and participate
	Notes:	

Week 16: Code.org, Course D, Lesson 9—Lesson Loops in Ice Age

Lesson overview:	**Purpose:** In this lesson, students will be learning more about loops and how to implement them in Blockly code. Using loops is an important skill in programming because manually repeating commands is tedious and inefficient. With these Code.org puzzles, students will learn to add instructions to existing loops, gather repeated code into loops, and recognize patterns that need to be repeated. This context-setting/skill-building lesson will quickly introduce students to loops. **Lesson:** ● Warm Up ○ Introduction to Loops ● Main Activity ○ Loops in Ice Age ● Wrap Up ○ Reflection ● Extended Learning ○ So Moving: Give the students pictures of actions or dance moves they can do. Have students arrange moves and add loops to choreograph their own dance. Share the dances with the rest of the class. ○ Connect It Back: Find some YouTube videos of popular dances that repeat themselves. Can your class find the loops? Try the same thing with songs.
Lesson links/resources:	Lesson 9: Loops in Ice Age
CS objectives addressed:	Students will be able to: ● Construct a program using structures that repeat areas of code ● Improve existing code by finding areas of repetition and moving them into looping structures Standards: ● **AP.1B.2**—Create program variables to store and modify data. ● **AP.1B.3**—Create programs that include sequences, events, loops, and conditionals. ● **AP.1B.4**—Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. ● **AP.1B.7**—Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.
Time needed:	**Total Time: 45-60 min** ● Warm Up **10 min** ● Main Activity **30 min** ● Wrap Up **5 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access

	• Feeling Faces—Emotion images • Pair Programming—Student video • Pause and Think Online—Video • Code.org account
Subject integrated:	Math
Other standards addressed:	**3.OA.9**—Identify arithmetic patterns (including patterns in the addition table or multiplication table) and explain them using properties of operations.
Vocabulary:	Loop: The action of doing something over and over Repeat: Do the same thing again
Notes:	

Week 17: Code.org, Course D, Lesson 10—Drawing Shapes With Loops

Lesson overview:	**Purpose:** This series highlights the power of loops with an array of puzzles meant to get students thinking about why repeat loops are superior to longhand. **Lesson:** ● Warm Up ○ Introduction ● Main Activity ○ Drawing Shapes with Loops ● Wrap Up ○ Reflection
Lesson links/resources:	Lesson 10: Drawing Shapes with Loops
CS objectives addressed:	Students will be able to: ● Differentiate between commands that need to be repeated in loops and commands that should be used on their own ● Identify the benefits of using a loop structure instead of manual repetition 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. ● **AP.1B.4**—Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. ● **AP.1B.8**—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 **15 min** ● Main Activity **30 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Code.org account
Subject integrated:	Math
Other standards addressed:	**3.OA.9**—Identify arithmetic patterns (including patterns in the addition table or multiplication table) and explain them using properties of operations.
Vocabulary:	Loop: The action of doing something over and over

Notes:

Week 18: Code.org, Course D, Lesson 11—Nested Loops in Maze

Lesson overview:	**Purpose:** In this introduction to nested loops, students will go outside of their comfort zone to create more efficient solutions to puzzles. In earlier puzzles, loops pushed students to recognize repetition. Here, students will learn to recognize patterns within repeated patterns to develop these nested loops. This stage starts by encouraging students to try to solve a puzzle where the code is irritating and too complex to write out the long way. After a video introduces nested loops, students are shown an example and asked to predict what will happen when a loop is put inside another loop. This progression leads to plenty of practice for students to solidify and build on their understanding of looping in programming. In this skill-building lesson, students will learn how to program a loop inside another loop. **Lesson:** ● Warm Up ○ Introduction ● Main Activity ○ Nested Loops in Maze ● Wrap Up ○ Reflection
Lesson links/resources:	Lesson 11: Nested Loops in Maze
CS objectives addressed:	Students will be able to: ● Break complex tasks into smaller repeatable sections ● Identify the benefits of using a loop structure instead of manual repetition ● Recognize large, repeated patterns as made from smaller, repeated patterns Standards ● **AP.1B.4**—Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. ● **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.8**—Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
Time needed:	**Total Time: 55-60 min** ● Warm Up **10 min** ● Main Activity **30 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Code.org account
Subject integrated:	Math

Other standards addressed:	**3.OA.9**—Identify arithmetic patterns (including patterns in the addition table or multiplication table) and explain them using properties of operations.
Vocabulary:	Command: An instruction for the computer. Many commands put together make up algorithms and computer programs. Loop: The action of doing something over and over
Notes:	This lesson can be divided into different sessions.

Week 19: Code.org, Course D, Lesson 12—Conditionals With Cards

Lesson overview: Coding Unplugged	**Purpose:** One of the best parts of teaching conditionals is that students already understand the concept from their everyday lives. This lesson merges computer science into the real world by building off their ability to tell if a condition is true or false. Students will learn to use if statements to declare when a certain command should be run, as well as if/else statements to declare when a command should be run and what to run otherwise. Students may not recognize the word conditionals, but most students will understand the idea of using if to make sure some action only occurs when it is supposed to. **Lesson:** ● Warm Up ○ Introduction ● Main Activity ○ Conditionals with Cards ● Wrap Up ○ Reflection ○ Reflection ○ Assessment ● Extended Learning ● Cross-Curricular Opportunity
Lesson links/resources:	Lesson 12: Conditionals with Cards
CS objectives addressed:	Students will be able to: ● Define circumstances when certain parts of a program should run and when they shouldn't ● Determine whether a conditional is met based on criteria ● Traverse a program and predict the outcome, given a set of input 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 **15 min** ● Main Activity **25 min** ● Wrap Up **10 min** ● Assessment **5 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Conditionals with Cards—Lesson in action video ● Conditionals with Cards—Assessment video ● Conditionals with Cards—Assessment answer key ● Conditionals with Cards Sample Program—Teacher prep guide ● Code.org account Students: ● Computer/tablet with internet access ● Conditionals with Cards—Unplugged video (Download) ● Conditionals with Cards—Assessment ● Code.org account

Subject integrated:	ELA Math
Other standards addressed:	ELA - **SL.3.3**—Ask and answer questions about information from a speaker, offering appropriate elaboration and detail. Math - **3.OA.7**—Fluently multiply and divide within 100, using strategies such as the relationship between multiplication and division or properties of operations.
Vocabulary:	Conditionals: Statements that only run under certain conditions Pseudocode: A plain language description of the steps in an algorithm
Notes:	

Week 20: Code.org, Course D, Lesson 12—Play the Deck With Conditionals

Lesson overview: Coding Unplugged	**Purpose:** Play the Deck with Conditionals is an activity aligned to Common Core ELA and math standards, written by our teacher community. Students will use a deck of playing cards to review basic multiplication and division problems as you apply the computer science concept of conditionals. **Lesson:** ● Class will watch the Conditionals with Cards Unplugged Activity video. ● Students will complete the Conditionals with Cards assessment activity. Note: Teachers can find an assessment teacher video explaining the answer key for this activity in the Conditionals with Cards lesson plan. ● The teacher will now incorporate multiplication and division skills into a whole class activity using the Play the Deck with Conditionals teacher slide deck. ○ The teacher may use a stack of playing cards to select cards, a virtual play deck or use the sample cards provided on the slide deck. ○ Students can write their answers on scrap paper or individual dry erase boards. ○ Answers are provided for the sample cards on the Play the Deck with Conditionals teacher slide deck in the speaker notes. ● As a class, now rewrite the conditional found on slide 2 of the Play the Deck with Conditionals teacher slide deck using different numbers. ● Have the class solve the math problems based on their selected playing cards using the new conditional they wrote in step 4. ● To wrap up, have students turn to a shoulder partner and discuss the following prompts: ○ What is a conditional? ○ How did you use a conditional today?
Lesson links/resources:	● Lesson 12: Conditionals with Cards ● Play the Deck with Conditionals
CS objectives addressed:	Students will be able to: ● Define circumstances when certain parts of a program should run and when they shouldn't ● Determine whether a conditional is met based on criteria ● Traverse a program and predict the outcome, given a set of input 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: Introductio**n 15 min** ● Main Activity: Play the Deck with Conditionals **30 min** ● Wrap Up: Reflection **10 min** ● Assessment **5 min**

Materials needed:	Teacher: - Computer - Projector/smartboard with sound - Code.org account Students: - Computer/tablet with internet access - Conditionals with Cards—Unplugged activity video - Conditionals with Cards—Assessment activity - Play the Deck with Conditionals—Teacher slide deck - Code.org account
Subject integrated:	ELA Math
Other standards addressed:	ELA - **SL.3.3**—Ask and answer questions about information from a speaker, offering appropriate elaboration and detail. Math - **3.OA.7**—Fluently multiply and divide within 100, using strategies such as the relationship between multiplication and division or properties of operations.
Vocabulary:	Conditionals: Statements that only run under certain conditions
Notes:	Repeat as needed or desired

Week 21: Code.org, Course D, Lesson 13—Looking Ahead With Minecraft

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:** ● Warm Up ○ Introduction ● Main Activity ○ Looking Ahead with Minecraft ● Wrap Up ○ Reflection ● Extended Learning ○ Allow students to continue to play Minecraft.
Lesson links/resources:	Lesson 13: Looking Ahead with Minecraft
CS objectives addressed:	Students will be able to: ● Define circumstances when certain parts of a program should run and when they shouldn't ● 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 **15 min** ● Main Activity **30 min** ● Wrap Up **5 min** ● Extended Learning **10 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Code.org account
Subject integrated:	ELA
Other standards addressed:	**RI.3.8**—Describe the logical connection between particular sentences and paragraphs in a text (e.g., comparison, cause/effect, first/second/third in a sequence).
Vocabulary:	Condition: Something a program checks to see if it is true before allowing an action Conditionals: Statements that only run under certain conditions
Notes:	

Week 22: Code.org, Course D, Lesson 14—If/Else With Bee

Lesson overview:	**Purpose:** After being introduced to conditionals in Conditionals With Cards, students will now practice using them in their programs. The "if/else" blocks will allow for a more flexible program. The bee will only collect nectar if there is a flower or make honey if there is a honeycomb. **Lesson:** ● Warm Up ○ Introduction ○ Preview of Online Puzzles ● Main Activity ○ If/Else With Bee ● Wrap Up ○ Reflection ● Extended Learning ○ True/False Tag
Lesson links/resources:	Lesson 14: If/Else with Bee
CS objectives addressed:	Students will be able to: ● Solve puzzles using a combination of looped sequences and conditionals ● Translate spoken language conditional statements into 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 **10 min** ● Main Activity **35 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Unplugged Blocks (Courses C-F)—Manipulatives ● Code.org account
Subject integrated:	ELA
Other standards addressed:	**W.3.3**—Write narratives to develop real or imagined experiences or events using effective technique, descriptive details, and clear event sequences.
Vocabulary:	Conditionals: Statements that only run under certain conditions
Notes:	This lesson can be divided into different sessions.

Week 23: Code.org, Course D, Lesson 15—Loops in Farmer

Lesson overview:	**Purpose:** As your students continue to deepen their knowledge of loops, they will come across problems where a command needs to be repeated, but it is unknown how many times it needs to be repeated. This is where while loops come in. In today's lesson, students will develop a beginner's understanding of condition-based loops and expand their knowledge of loops in general. **Lesson:** ● Warm Up ○ Introduction ○ Preview of Online Puzzles ● Main Activity ○ While Loops in Farmer ● Wrap Up ○ Reflection
Lesson links/resources:	Lesson 15: Loops in Farmer
CS objectives addressed:	Students will be able to: ● Distinguish between loops that repeat a fixed number of times and loops that repeat as long as a condition is true ● Use a while loop to create programs that can solve problems with unknown values 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 **10 min** ● Main Activity **35 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Conditionals with Cards sample program—Teacher prep guide ● Code.org account Students: ● Computer/tablet with internet access ● Unplugged Blocks (Courses C-F)—Manipulatives ● Code.org account
Subject integrated:	Math
Other standards addressed:	**3.OA.9**—Identify arithmetic patterns (including patterns in the addition table or multiplication table) and explain them using properties of operations.
Vocabulary:	Condition: Something a program checks to see if it is true before allowing an action Loop: The action of doing something over and over While loop: A loop that continues to repeat while a condition is true

Notes:

Week 24: Code.org, Course D, Lesson 16—Until Loops in Maze

Lesson overview:	**Purpose:** This set of puzzles will work to solidify and build on the knowledge of loops by adding the until conditional. By pairing these concepts together, students will be able to explore the potential for creating complex and innovative programs. **Lesson:** ● Warm Up ○ Introduction ● Main Activity ○ Until Loops in Maze ● Wrap Up ○ Reflection
Lesson links/resources:	Lesson 16: Until Loops in Maze
CS objectives addressed:	Students will be able to: ● Build programs with the understanding of multiple strategies to implement conditionals ● Translate spoken language conditional statements and loops into 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 **10 min** ● Main Activity **30-35 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Code.org account
Subject integrated:	Math
Other standards addressed:	**3.OA.9**—Identify arithmetic patterns (including patterns in the addition table or multiplication table) and explain them using properties of operations.
Vocabulary:	Condition: Something a program checks to see if it is true before allowing an action Conditionals: Statements that only run under certain conditions Coop: The action of doing something over and over Until: A command that tells you to do something only up to the point that something becomes true
Notes:	This lesson can be divided into different sessions.

Week 25: Dialogue

Lesson overview:

Purpose:

Students will learn how to use block coding to create a digital story. This is the beginning of a series of 10 lessons.

Lesson:

- Students watch videos and create a "Dialogue" 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 storytelling project in Scratch. Review and model comma usage and quotation marks.

Student Tasks: Open Scratch Account

1. Choose a story starter.

Story starters:

- I wouldn't do that if I were you.
- It was br/ave of you to come here today.
- They lied to you.
- You won!
- That was a dangerous move that you pulled back there.

2. Add two sprites.
3. Make the sprites face each other.
4. Add a backdrop.
5. Make the sprite say the starter phrase you selected.
6. Make the second sprite talk as well.
7. Start the code for each sprite when the green flag is clicked.
8. Make the second sprite wait for the first to finish talking before starting its dialogue.
9. Continue to build your dialogue.
10. Watch some videos to learn about add-ons.
    Add a Title Screen
    Costume Animation 1 of 2
    Costume Animation 2 of 2
    Adding Motion (challenge)
    Add a Third Character
    Add a Second Scene to your Story (Challenge)

11. Follow your teacher’s directions to share or save the project.

- Check in with students as they complete their projects.
- Encourage students to also share their projects.
- Discuss the lesson and facilitate a brief discussion about what students learned and experienced.
    - Question 1: What was your favorite part of this lesson?
    - Question 2: What was challenging about sequencing your code?
    - Question 3: What was most interesting about your story or a neighbor's story?
    - Dialogue Example Project

If you get stuck, review the Dialogue Solution Sheet.

Lesson links/resources:

- Storytelling Instructional Slides
- Dialogue Example Project
- If you get stuck, review the Dialogue Solution Sheet.

CS objectives addressed:

Students will be able to:

- Develop their computer science skills using Scratch
- Create a program using Scratch

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.12**—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:	**<u>Total Time: </u>50-60 min** ● Introduction (Survey) **10 min** ● Dialogue Videos/Activity **30-40 min** ● Wrap Up **10 min**
Materials needed:	Teacher: Teacher: ● Computer ● Projector/smartboard with sound ● Storytelling Instructional Slides ● Dialogue Example Project ● If you get stuck, review the Dialogue Solution Sheet. ● Scratch Account Note: There are additional instructional videos on creating stories using scratch on YouTube. Students: ● Computer/tablet with internet access ● Scratch Account
Subject integrated:	ELA
Other standards addressed:	● **L.3.2**—Use commas and quotation marks in dialogue. ● **RL.3.2**—Recount stories, including fables, folktales, and myths from diverse cultures; determine the central message, lesson, or moral and explain how it is conveyed through key details in the text.
Vocabulary:	<u>Conditionals</u>: Statements that only run under certain conditions <u>Loop</u>: The action of doing something over and over <u>Event</u>: An action that causes something to happen <u>Sequence</u>: An action that causes something to happen
Notes:	This lesson can be divided into different sessions. →Teachers will need to create free teacher and/or student accounts (when applicable) at https://scratch.mit.edu This lesson was recreated using resources from CS First which is no longer available online.

Week 26: Check It Out

Lesson overview:	**Purpose:** Students tell a story using Scratch where a character walks through a scene describing what they see. **Lesson:** ● Have students log in to the scratch account. • Videos: Introduction Unexpected Encounter (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. ● Students watch videos and create a "Check It Out" project in Scratch. ● Check in with students as they watch the videos and complete their projects. ● Students choose add-ons to enhance their project. ● Discuss the lesson and facilitate a brief discussion about what students learned and experienced. ○ Question 1: What was your favorite part of this lesson? ○ Question 2: What story did you tell? ○ Question 3: What blocks did you use, and what did they do?
Lesson links/resources:	● Storytelling Slide ● Starter Project One ● Starter Project Two ● Starter Project Three ● If you get stuck, review the Solution Sheet.
CS objectives addressed:	Students will be able to: ● Develop their computer science skills using Scratch ● Create a program using Scratch 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.12**—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: 50-60 min** ● Introduction (Survey) **10 min** ● Check It Out Videos/Activity **30-40 min** ● Wrap Up **10 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Starter Project One ● Starter Project Two ● Starter Project Three ● If you get stuck, review the Solution Sheet. ● Scratch Account Students:

	- Computer/tablet with internet access - Scratch Account
Subject integrated:	ELA
Other standards addressed:	**RL.3.3**—Describe characters in a story (e.g., their traits, motivations, or feelings) and explain how their actions contribute to the sequence of events.
Vocabulary:	Sequencing: The specific order in which instructions are performed in an algorithm
Notes:	This lesson can be divided into different sessions. This lesson was recreated using resources from CS First.

Week 27: Setting

Lesson overview:

Purpose:
Students create a dynamic stormy day setting, complete with rain and lightning. After developing the setting, students program a stormy day story.

Lesson:

- Recap the last lesson and have students log in to their Scratch account.

Videos: Introduction to setting   Making it Rain

- (Use smartboard to show videos in attached slide or assign the Storytelling Lesson Plan Slide to students in Google classroom for them to view independently.) *Youtube has some additional instructional videos if needed.

- Check in with students as they watch the videos and complete their projects.
    - Video 2: Common problem: The rain moves off the stage and is difficult to retrieve.
        - If a sprite is ever out of view (preventing you from dragging it), click a "go to" block with x and y values of 0 (in the motion menu). This will move the sprite to the middle of the stage.
- Common problem: A student wants to reorder the layering of sprites (either the earth sprite is on top of the rain, or the rain sprite is on top of the earth).
    - To set a sprite to the front stage position, select the looks menu and click "go to front."
    - Video 3: Common problem: The backdrop is stuck on the lightning flash.
        - The program ended with the lightning backdrop shown. This can be fixed by placing a "change backdrop to night sky" block after a "when flag clicked" block. Or simply click the stage, select the backdrops tab, and select the night sky.
    - Video 4: Students will need to place a loop within a loop. The ordering of these instructions can be confusing, especially to novice programmers. If you notice students have difficulty sequencing these instructions, ask them to read the inner loop first, then the outer loop. The inner loop should make the lightning flash by changing backdrops. The outer loop should make that lightning flash at random times throughout the program.
    - Video 5: This video asks students to create a story on top of this stormy day setting. If they are having difficulty, suggest they start by using "say" blocks and see where the story takes them.
- Students choose add-ons to enhance their project.

Lightning Reaction
Drawing and Animating a Lightning Bolt
Sun Comes Out
Playing Thunder
Earthquake
Code A Rainbow

- When there are five minutes left in class, instruct students to watch the wrap up video.

	● Encourage students to also show their projects to a neighbor/classmate. ● Discuss the lesson and facilitate a brief discussion about what students learned and experienced. ○ Question 1: What was the story you built in the stormy day setting? ○ Question 2: What does it mean if something happens randomly? ○ Question 3: What do loops do? (Answer: repeat instructions)
Lesson links/resources:	● Storytelling Lesson Plan Slide ● Setting Starter Project ● Setting Example Project ● If you get stuck, review the Solution Sheet..
CS objectives addressed:	Students will be able to: ● Develop their computer science skills using Scratch ● Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 50-60 min** ● Introduction (Survey) **10 min** ● Setting Videos/Activity **30-40 min** ● Wrap Up **10 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Storytelling Lesson Plan Slide ● Setting Starter Project ● Setting Example Project ● If you get stuck, review the Solution Sheet. ● Scratch Account Students: ● Computer/tablet with internet access ● Scratch Account
Subject integrated:	ELA
Other standards addressed:	● **RL.3.3**—Describe characters in a story (e.g., their traits, motivations, or feelings) and explain how their actions contribute to the sequence of events. ● **RL.3.5**—Refer to parts of stories, dramas, and poems when writing or speaking about a text, using terms such as chapter, scene, and stanza; describe how each successive part builds on earlier sections.

Vocabulary:	Randomness: Any collection of data or information with no determined order Loops: Blocks of code that repeat over and over
Notes:	This lesson can be divided into different sessions. This lesson was recreated using resources from CS First.

Week 28: Lesson 1 of 2: Premise

Lesson overview:	**Purpose:** **Premise: (1 of 2 days)** Students build a story around one of four premises. **Lesson:** ● Recap the last lesson and have students log in to their Scratch account. ○ Students watch Introduction to Premise and create a "Premise" project in Scratch. Students will choose one starter project with which to build their story. ● Sprite vs Nature Starter ● Sprite vs Self Starter ● Sprite vs Society Starter ● Sprite vs Sprite Starter ● (Use smartboard to show videos in attached slide or assign the Storytelling Plan Slide Lesson to students in Google classroom for them to view independently.) *Youtube has some additional instructional videos if needed. ● Watch Introduction to Modules. Select modules that interest you! ● Check in with students as they watch the videos and complete their projects. ○ Students choose add-ons to enhance their project. Watch Ending video and add some add-ons to your project. ● Escape ● Dynamic Dialogue ● Between the Scenes ● DeusExMachina ● The End ● When there are five minutes left in class, instruct students watch the Wrapping Up video. Your students' projects are automatically shared with your teacher account. Encourage students to also show their projects to a neighbor/classmate. ● Discuss the lesson and facilitate a brief discussion about what students learned and experienced. ○ Question 1: What was your favorite part of this lesson? ○ Question 2: What did you like about your story once you finished?
Lesson links/resources:	● Storytelling Plan Slide Lesson ● Sprite vs Nature Starter ● Sprite vs Self Starter ● Sprite vs Society Starter ● Sprite vs Sprite Starter ● Premise Example Project ● If you get stuck, review the Solution Sheet.
CS objectives addressed:	Students will be able to: ● Develop their computer science skills using Scratch ● Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 50-60 min** - Introduction (Survey) **10 min** - Premise Videos/Activity **40-50 min**
Materials needed:	Teacher: - Computer - Projector/smartboard with sound - Storytelling Plan Slide Lesson - Sprite vs Nature Starter - Sprite vs Self Starter - Sprite vs Society Starter - Sprite vs Sprite Starter - Premise Example Project - Solution Sheet. - Scratch Account Students: - Computer/tablet with internet access - Scratch Account
Subject integrated:	ELA
Other standards addressed:	- **RL.3.3**—Describe characters in a story (e.g., their traits, motivations, or feelings) and explain how their actions contribute to the sequence of events. - **RL.3.5**—Refer to parts of stories, dramas, and poems when writing or speaking about a text, using terms such as chapter, scene, and stanza; describe how each successive part builds on earlier sections.
Vocabulary:	Modularity: Adding many different components to a project
Notes:	This lesson can be divided into different sessions. This lesson was recreated using resources from CS First.

Week 29: Lesson 2 of 2: Premise

Lesson overview:	**Purpose:** **Premise: (1 of 2 days)** Students build a story around one of four premises. **Lesson:** • Recap the last lesson and have students log in to their Scratch account. ○ Students watch Introduction to Premise and create a "Premise" project in Scratch. Students will choose one starter project with which to build their story. • Sprite vs Nature Starter • Sprite vs Self Starter • Sprite vs Society Starter • Sprite vs Sprite Starter • (Use smartboard to show videos in attached slide or assign the Storytelling Plan Slide Lesson to students in Google classroom for them to view independently.) *Youtube has some additional instructional videos if needed. • Watch Introduction to Modules. Select modules that interest you! • Check in with students as they watch the videos and complete their projects. ○ Students choose add-ons to enhance their project. Watch Ending video and add some add-ons to your project. • Escape • Dynamic Dialogue • Between the Scenes • DeusExMachina • The End • When there are five minutes left in class, instruct students watch the Wrapping Up video. Your students' projects are automatically shared with your teacher account. Encourage students to also show their projects to a neighbor/classmate. • Discuss the lesson and facilitate a brief discussion about what students learned and experienced. ○ Question 1: What was your favorite part of this lesson? ○ Question 2: What did you like about your story once you finished?
Lesson links/resources:	• Storytelling Lesson Plan—CS First • Premise Example Project • Solution Sheet
CS objectives addressed:	Students will be able to: • Develop their computer science skills using scratch • Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 50-60 min** ● Premise Add-ons **40-50 min** ● Wrap Up **10 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Storytelling Plan Slide Lesson ● Sprite vs Nature Starter ● Sprite vs Self Starter ● Sprite vs Society Starter ● Sprite vs Sprite Starter ● Premise Example Project ● Solution Sheet. ● Scratch Account Students: ● Computer/tablet with internet access ● Scratch Account
Subject integrated:	ELA
Other standards addressed:	● **RL.3.3**—Describe characters in a story (e.g., their traits, motivations, or feelings) and explain how their actions contribute to the sequence of events. ● **RL.3.5**—Refer to parts of stories, dramas, and poems when writing or speaking about a text, using terms such as chapter, scene, and stanza; describe how each successive part builds on earlier sections
Vocabulary:	Modularity: Adding many different components to a project
Notes:	This lesson was recreated using resources from CS First.

Week 30: Lesson 1 of 2: Characterization

Lesson overview:

Purpose:
In this lesson, students will program a project that describes in detail a character in a story.

Lesson:

- Recap the last lesson and have students log in to their Scratch accounts.
- Students watch videos and create a "Characterization" project in Scratch.
- (Use smartboard to show videos in attached slide or assign the Storytelling Plan Slide Lesson to students in Google classroom for them to view independently.) *Youtube has some additional instructional videos if needed.
- Video 1: Introduction
- Check in with students as they watch the videos and complete their projects.
    - Video 2: Students may get stuck planning their story. Encourage them to start with whatever ideas they have and that there are no wrong answers.
    - Video 3: Students may get confused as to which blocks should go with each sprite. The narrator (cat) sprite should have a lot of "say" blocks and "broadcast" blocks. No code is added to the character sprite during this video.
    - Video 4: All code from this video will be for the character sprite. At the end of this video, the code won't run in the project unless students click on it. The code will be integrated into the story in the next video.
    - Video 5: Common problem: Make sure the narrator broadcasts the message with a "broadcast" block and the character sprite receives it with a "when I receive" block.
        - If the character doesn't walk, students may have forgotten to change the message being sent and received. Make sure they have created a separate message for "walking."
- Video: Students choose add-ons to enhance their project.
- When there are five minutes left in class, instruct students watch the wrap up video.
- Encourage students to also show their projects to a neighbor/classmate.
- Discuss the lesson and facilitate a brief discussion about what students learned and experienced.
- Question 1: What was your favorite part of this lesson?
- Question 2: How did you use "broadcast" in your story?

Lesson links/resources:

- Storytelling Plan Slide Lesson
- Dino Starter Project
- Horse Starter Project
- Fairy Tale Starter Project
- Soccer Starter Project
- Bird Started Project
- Characterization Example Project
- If you get stuck, review the Solution Sheet.

CS objectives addressed:

Students will be able to:

- Develop their computer science skills using Scratch

	● Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 50-60 min** ● Introduction (Survey) **10 min** ● Characterization Vide ● os/Activity **40-50 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Storytelling Plan Slide Lesson ● Dino Starter Project ● Horse Starter Project ● Fairy Tale Starter Project ● Soccer Starter Project ● Bird Started Project ● Characterization Example Project ● If you get stuck, review the Solution Sheet. ● Scratch Account Students: ● Computer/tablet with internet access ● Scratch Account
Subject integrated:	ELA
Other standards addressed:	**RL.3.3**—Describe characters in a story (e.g., their traits, motivations, or feelings) and explain how their actions contribute to the sequence of events.
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:	This lesson was recreated using resources from CS First.

Week 31: Lesson 2 of 2: Characterization

Lesson overview:	**Purpose:** In this lesson, students will program a project that describes in detail a character in a story. **Lesson:** ● Recap the last lesson and have students log in to their Scratch accounts. ● Students watch videos and create a "Characterization" project in Scratch. ● (Use smartboard to show videos in attached slide or assign the Storytelling Plan Slide Lesson to students in Google classroom for them to view independently.) *Youtube has some additional instructional videos if needed. ● Video 1: Introduction ● Check in with students as they watch the videos and complete their projects. ○ Video 2: Students may get stuck planning their story. Encourage them to start with whatever ideas they have and that there are no wrong answers. ○ Video 3: Students may get confused as to which blocks should go with each sprite. The narrator (cat) sprite should have a lot of "say" blocks and "broadcast" blocks. No code is added to the character sprite during this video. ○ Video 4: All code from this video will be for the character sprite. At the end of this video, the code won't run in the project unless students click on it. The code will be integrated into the story in the next video. ○ Video 5: Common problem: Make sure the narrator broadcasts the message with a "broadcast" block and the character sprite receives it with a "when I receive" block. ■ If the character doesn't walk, students may have forgotten to change the message being sent and received. Make sure they have created a separate message for "walking." ● Video: Students choose add-ons to enhance their project. ● When there are five minutes left in class, instruct students watch the wrap up video. ● Encourage students to also show their projects to a neighbor/classmate. ● Discuss the lesson and facilitate a brief discussion about what students learned and experienced. ● Question 1: What was your favorite part of this lesson? ○ Question 2: How did you use "broadcast" in your story?
Lesson links/resources:	● Storytelling Plan Slide Lesson ● Dino Starter Project ● Horse Starter Project ● Fairy Tale Starter Project ● Soccer Starter Project ● Bird Started Project ● Characterization Example Project ● If you get stuck, review the Solution Sheet
CS objectives addressed:	Students will be able to: ● Develop their computer science skills using Scratch

	• Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 50-60 min** • Characterization Videos/Activity/Add-ons **30-40 min** • Wrap-Up **10 min**
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Storytelling Plan Slide Lesson • Dino Starter Project • Horse Starter Project • Fairy Tale Starter Project • Soccer Starter Project • Bird Started Project • Characterization Example Project • If you get stuck, review the Solution Sheet. • Scratch Account Students: • Computer/tablet with internet access • Scratch Account
Subject integrated:	ELA
Other standards addressed:	**RL.3.3**—Describe characters in a story (e.g., their traits, motivations, or feelings) and explain how their actions contribute to the sequence of events.
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:	This lesson was recreated using resources from CS First.

Week 32: Interactive Storytelling

Lesson overview:

Coding

Purpose:
Students create a story in which the audience can make a decision.

Lesson:

- Recap the last lesson and have students log in to their Scratch Account.
- Students watch videos and create an "Interactive Storytelling" project.
- (Use smartboard to show videos in attached slide or assign the Storytelling Plan Slide Lesson to students in Google classroom for them to view independently.) *Youtube has some additional instructional videos if needed.
- Check in with students as they watch the videos and complete their projects. Watch Introduction to Interactive Stories.
    - Video 2: Show students that what they type in the box at the bottom changes the value of "answer."
    - If they can't see the value of "answer" in the top left corner of their screen, go to the sensing menu and check the box next to "answer."
    - Video 3: Possible problem: If the condition contains spaces that aren't in the answer, the program won't work. For example, " yes" does not equal "yes."
    - Possible problem: The "switch backdrop to" block has the wrong backdrop selected. Have students compare the name of the backdrop to the backdrop selected in the dropdown.

1. Video 4: Add a story around the secret door decision.

- Video 5: Students choose add-ons to enhance their project.
- What's the Password
- Find the Secret Door
- Add Another Character
- Add Another Scene

- When there are five minutes left in class, instruct students to watch the Wrap-Up video.
- Encourage students to also show their projects to a neighbor/classmate.
- Discuss the lesson and facilitate a brief discussion about what students learned and experienced.
    - Question 1: What was your favorite part of this lesson?
    - Question 2: How did you use an if/else statement in your project?
    - Question 3: What did you program in each world you created?

Lesson links/resources:

- Interactive Storytelling Starter Project
- Interactive Storytelling Example Project
- If you get stuck, review the Interactive Storytelling Solution Sheet

CS objectives addressed:	Students will be able to: ● Develop their computer science skills using Scratch ● Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 50-60 min** ● Introduction (Survey) **10 min** ● Storytelling Videos/Activity **30-40 min** ● Wrap Up **10 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Storytelling Plan Slide Lesson ● Interactive Storytelling Starter Project ● Interactive Storytelling Example Project ● If you get stuck, review the Solution Sheet. ● Scratch Account Students: ● Computer/tablet with internet access ● Scratch Account
Subject integrated:	ELA
Other standards addressed:	**RL.3.3**—Describe characters in a story (e.g., their traits, motivations, or feelings) and explain how their actions contribute to the sequence of events.
Vocabulary:	Conditionals: Statements that only run under certain conditions
Notes:	This lesson was recreated using resources from CS First.

Week 33: Personal Narrative

Lesson overview:	**Purpose:** Students create a personal narrative based on one of three story starters. **Lesson:** • Recap the last lesson and have students log in to their Scratch Account. Students watch videos and create a "Personal Narrative" project in Scratch. • (Use smartboard to show videos in attached slide or assign the Storytelling Plan Slide Lesson to students in Google classroom for them to view independently.) *Youtube has some additional instructional videos if needed. • Check in with students as they watch the videos and complete their projects. • Video 1: Choose a premise for your personal narrative, and program it: • My perfect day.. • My life if I were famous... • If I had a superpower... • Video 2: Choose an Add-On, and "watch" to learn how to build it. • Write A Song • Add Credits Scene • Easter Egg • Add A Second Scene • User-Controlled Decision • When there are five minutes left in class, instruct students to watch the Wrap Up video. • Encourage students to also show their projects to a neighbor/classmate. • Discuss the lesson and facilitate a brief discussion about what students learned and experienced. • Encourage students to show their projects to a neighbor/classmate. • Discuss the lesson and facilitate a brief discussion about what students learned and experienced. ○ Question 1: What did you program your story to do? ○ Question 2: What types of blocks did you use in your story? ○ Question 3: Did you use any blocks for this lesson that you haven't used before?
Lesson links/resources:	• Storytelling Plan Slide Lesson • Personal Narrative Example Project • If you get stuck, review the Solution Sheet.
CS objectives addressed:	Students will be able to: • Develop their computer science skills using Scratch • Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 50-60 min** - Introduction (Survey) **10 min** - Personal Narrative Videos/Activity **30-40 min** - Wrap Up **10 min**
Materials needed:	Teacher: - Computer - Projector/smartboard with sound - Storytelling Plan Slide Lesson - Personal Narrative Example Project - If you get stuck, review the Solution Sheet. - Scratch Account Students: - Computer/tablet with internet access - Scratch Account
Subject integrated:	ELA
Other standards addressed:	**W.3.3**—Write narratives to develop real or imagined experiences or events using effective technique, descriptive details, and clear event sequences
Vocabulary:	Conditionals: Statements that only run under certain conditions Loops: Blocks of code that repeat over and over Sequencing: The specific order in which instructions are performed in an algorithm
Notes:	This lesson was recreated using resources from CS First.

Week 34: Story of an Innovation

Lesson overview:	**Purpose:** Students tell the story of an innovation that they create. **Lesson:** • Recap the last lesson and have students log in to their Scratch Accounts. • Students watch videos and create a "**Your Innovation Story Project.**" in Scratch. • (Use smartboard to show videos in attached slide or assign the Storytelling Plan Slide Lesson to students in Google classroom for them to view independently.) *Youtube has some additional instructional videos if needed. • Video 1: Your Innovation Story • Check in with students as they watch the videos and complete their projects. ○ Video 2: Look for signs that students haven't selected an innovation (e.g., scribbling in the paint editor, dragging out many unconnected blocks, etc.). • Video: Students choose add-ons to enhance their project. • Scene Change • Protect Your Idea • Building Volume Add Interactivity When there are five minutes left in class, instruct students to watch the Wrap Up video. • Your students' projects are automatically shared with your teacher account. Encourage students to also show their projects to a neighbor/classmate. • Discuss the lesson and facilitate a brief discussion about what students learned and experienced. ○ Question 1: What was your favorite program that you made in this class and why? ○ Question 2: What was the most surprising thing that you were able to do? ○ Question 3: What was the most important thing that you learned about computer science? ○ Question 4: What do computer scientists do? ○ Question 5: Does anyone have any questions for me about what we've worked on in this class or about computer science in general?
Lesson links/resources:	• Storytelling Plan Slide Lesson • **Your Innovation Story Project.** • Your Innovation Story Example Project • If you get stuck, review the Solution Sheet..
CS objectives	Students will be able to:

addressed:	- Develop their computer science skills using Scratch - Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 50-60 min** - Introduction (Survey) **10 min** - Your Story of Innovation Videos/Activity **30-40 min** - Wrap Up **10 min**
Materials needed:	Teacher: - Computer - Projector/smartboard with sound - Storytelling Plan Slide Lesson - **Your Innovation Story Project.** - Your Innovation Story Example Project If you get stuck, review the Solution Sheet. - Scratch Account Students: - Computer/tablet with internet access - Scratch Account
Subject integrated:	ELA
Other standards addressed:	**W.3.2**—Write informative/explanatory texts to examine a topic and convey ideas and information clearly.
Vocabulary:	Innovation: A new method, idea, or product
Notes:	This lesson was recreated using resources from CS First.

Week 35: Ecosystems in Sprite Lab

Lesson overview: Coding	**Purpose:** This project asks students to use their coding skills to develop a model of a system. Students might model ecosystems or interactions between the geosphere, biosphere, hydrosphere, and/or atmosphere. **Lesson:** ● Students might model ecosystems or interactions between the geosphere, biosphere, hydrosphere, and/or atmosphere. ● Warm Up ○ Predict ● Main Activity ○ Run ○ Investigate ○ Modify ○ Make ● Wrap Up ○ Reflection
Lesson links/resources:	CS Connections: Ecosystems in Sprite Lab
CS objectives addressed:	Students will be able to: ● Develop their computer science skills using Scratch ● Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 70 min** ● Warm Up **10 min** ● Main Activity **45 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Code.org account
Subject integrated:	Science
Other standards addressed:	**E.3.9**—Students will demonstrate an understanding of how the Earth's systems (i.e., geosphere, hydrosphere, atmosphere, and biosphere) interact in multiple ways to affect Earth's surface materials and processes.
Vocabulary:	Sprite: A graphic on the screen with a location, size, and appearance

	Costume: An image that gives a sprite its appearance Event: An action that causes something to happen
Notes:	Ideas for adapting project

Week 36: Life Cycles in Sprite Lab

Lesson overview:	**Purpose:** This project asks students to use their coding skills to develop a model of a scientific concept learned in class. Students can model anything from life cycles to the transfer of energy or information. **Lesson:** ● Introduction ○ Predictions made about the program ● Main Activity ○ Observe the program and compare predictions. ○ Create an individual sprite lab project related to any scientific concept. ● Wrap Up ○ Reflection ○ Discussion
Lesson links/resources:	CS Connections: Life Cycles in Sprite Lab
CS objectives addressed:	Students will be able to: ● Develop their computer science skills using Scratch ● Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 70 min**. ● Warm Up **10 min** ● Main Activity **45 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Code.org account
Subject integrated:	Science
Other standards addressed:	**3.LS1.1**—Develop models to describe that organism have unique and diverse life cycles but all have in common birth, growth, reproduction, and death. (Next Gen Science Standards)
Vocabulary:	Sprite: A graphic on the screen with a location, size, and appearance

	Costume: An image that gives a sprite its appearance Event: An action that causes something to happen
Notes:	Adapting ideas for this project

Week 37: Character Study in Sprite Lab

Lesson overview:	**Purpose:** This project asks students to use their coding skills to recount stories and describe characters. Students could apply this skill to any story, folktale, book, or fable. **Lesson:** ● Introductory Tutorial for Newbies—Hello World ● Pre-Lesson (Optional)—Read the story "The Tortoise and the Hare" ● Getting Started—Video ○ Creating a class section ○ Introduction to PRIMM on code.org (optional if already completed) ● Starter Program—Character Study ○ Lesson Slides—Character Study ● Make/Create a Sprite Lab Project
Lesson links/resources:	CS Connections: Character Study in Sprite Lab
CS objectives addressed:	Students will be able to: ● Develop their computer science skills using Scratch ● Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 70 min** ● Warm Up **10 min** ● Main Activity **45 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Code.org account
Subject integrated:	ELA
Other standards addressed:	● **RL.3.3**—Describe characters in a story (e.g., their traits, motivations, or feelings) and explain how their actions contribute to the sequence of events. ● **RL.3.5**—Refer to parts of stories, dramas, and poems when writing or speaking about a text, using terms such as chapter, scene, and stanza; describe how each successive part builds on earlier sections.

Vocabulary:	Sprite: A graphic on the screen with a location, size, and appearance Costume: An image that gives a sprite its appearance Event: An action that causes something to happen
Notes:	Adapting ideas for this project

Week 38: Story Morals in Sprite Lab

Lesson overview:	**Purpose:** This project asks students to use their coding skills to recount stories and determine their morals. Students could apply this skill to any story, folktale, book, or fable. **Lesson:** ● Introductory Tutorial for Newbies—Hello World ● Pre-Lesson (Optional)—Read the text "Stone Soup" ● Getting Started—Video ○ Creating a class section ○ Introduction to PRIMM on code.org (optional if already completed) ● Starter Program—Story Morals ○ Lesson Slides—Story Morals ● Make/Create a Sprite Lab Project
Lesson links/resources:	Story Morals in Sprite Lab
CS objectives addressed:	Students will be able to: ● Develop their computer science skills using Scratch ● Use Scratch to create a program using sequences, events, loops, and conditionals 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.12**—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: 70 min** ● Warm Up **10 min** ● Main Activity **45 min** ● Wrap Up **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Code.org account Students: ● Computer/tablet with internet access ● Code.org account
Subject integrated:	ELA
Other standards addressed:	● **RL.3.2**—Recount stories, including fables, folktales, and myths from diverse cultures; determine the central message, lesson, or moral and explain how it is conveyed through key details in the text. ● **RL.3.5**—Refer to parts of stories, dramas, and poems when writing or speaking about a text, using terms such as chapter, scene, and stanza; describe how each successive part builds on earlier sections

Vocabulary:	Sprite: A graphic on the screen with a location, size, and appearance Costume: An image that gives a sprite its appearance Event: An action that causes something to happen
Notes:	Adapting ideas for this project

Week 39: Find the Area/Robot Game

Lesson overview: Coding Robotics Unplugged	**Purpose:** In this lesson, students will play the Code & Go Robot Mouse Game. They will choose a rectangle and find the missing side length or the total area. Students will then program the mouse to go to the correct answer on the maze. The teacher will need to make index cards or post-its with various area problems in advance, as well as answers to these problems. The answers will be the grid. Lesson: ● Warm Up ● The teacher will review the concept of area and introduce today's activity. ● Main Activity ○ Robot Game: The students will draw a card with an area problem. Students will identify the area and then program their Code & Go Robot Mouse or similar codable robot to go to that area on a grid. (If you do not have a codable robot, the students can act as the robot.) Students will switch roles until everyone has had an opportunity to participate in given roles. ● Wrap Up ○ Students will explain how they programmed the robot mouse through discussion. ● Online Keyboarding Practice
Lesson links/resources:	Robot Game
CS objectives addressed:	Students will be able to: ● Program a robot mouse to identify the area of a rectangle ● Test and debug a program to achieve a specific goal Standards: ● **AP.1B.8**—Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (Program Development) ● **AP.1B.9**—Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. ● **AP.1B.10**—Describe choices made during program development using code comments, presentations, and demonstrations.
Time needed:	**Total Time: 60 min** ● Warm Up **5 min** ● Main Activity **40 min** ● Wrap Up **5 min** ● Online Keyboarding Practice **10 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Robot Game Students: ● Computer/tablet with internet access ● Robot (Some examples include: Code and Go Mouse, Botley, Dash)

	- Index cards/sticky notes with a variety of rectangles labeled with side lengths, missing side lengths, or total area. - Answer key - Astro Bubbles Keyboarding Game
Subject integrated:	Math
Other standards addressed:	**3.MD.7**—Relate area to the operations of multiplication and addition.
Vocabulary:	Area: Space taken up by a rectangular shape Debugger: The process of removing existing errors Programmer: The individual who sets the instructions on a robot
Notes:	You could also use perimeter cards as well.

Week 40: Character Traits Robot Game

Lesson overview: Coding Robotics Unplugged	**Purpose:** In this lesson, students will participate in a robot game using the Code & Go Robot Mouse or similar codable robot. (If you do not have a codable robot, the students can act as the robot.) Students will identify character traits throughout the book and program their mouse to locate the correct trait for a given character. Then, students will program the mouse to go to the correct answer on the maze. ● Allow students opportunities to explain their reasoning. ● This could be used as a whole group or small group/center activity. ● Teachers will need to make cards with various character traits in advance to be used as a grid. Lesson: ● Warm Up ○ Generate a list of character traits. ● Main Activity ○ Read or listen online to the book *The Day the Crayons Quit* by Drew Daywait. ○ Play the Robot Game: The teacher will name a character/crayon from the story or have a stack of character cards for students to draw from. ○ Students will identify the character trait for that particular character and then program the Code & Go Robot Mouse (or similar robot) to reach that desired answer. Students will switch roles until everyone has had an opportunity to participate in given roles. ● Wrap Up ○ Students will explain how they programmed the robot mouse through discussion.
Lesson links/resources:	● *The Day the Crayons Quit* ● Robot Game
CS objectives addressed:	Students will be able to: ● Program a robot mouse to identify character traits ● Test and debug a program to achieve a specific goal Standards: ● **AP.1B.8**—Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (Program Development) ● **AP.1B.9**—Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. ● **AP.1B.10**—Describe choices made during program development using code comments, presentations, and demonstrations.
Time needed:	**Total Time: 50 min** ● Warm Up **5 min** ● Main Activity **40 min** ● Wrap Up **5 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Robot (Some examples include: Code and Go Mouse, Botley, Dash)

	- Index cards or post-its with various character traits (e.g., honest, grumpy, happy, etc.) - *The Day the Crayons Quit* by Drew Daywait - Answer key
Subject integrated:	ELA
Other standards addressed:	**RL.3.3**—Describe characters in a story (e.g., their traits, motivations or feelings) and explain how their actions contribute to the sequence of events.
Vocabulary:	Debugger: The process of removing existing errors Programmer: The individual who sets the instructions on a robot
Notes:	This lesson can be broken down into more than one session with additional books within Lexile range. *The Day the Crayons Quit* has a Lexile of 730.

Week 41: The Role of the Three Branches of Government

Lesson overview: Coding Robotics Unplugged	**Purpose:** The students will participate in a Robot Game using the Code & Go Robot Mouse or similar codable robot. (If you do not have a codable robot, the students can act as the robot.) Students will identify the distribution of power in the three branches of government. Then, students will use the order pairs to program the mouse to move to the correct answer regarding the powers of government on the maze. **Lesson:** ● Warm Up ○ Review the structure and function of various levels of government (e.g., local, state, federal). ● Main Activity ○ Using ordered pairs, program the mouse to select correct answers in the robot game related to power of the three branches of government. ● Wrap Up ○ Reflect and discuss how students programmed the mouse to select the correct answers within the game.
Lesson links/resources:	● Three Branches of Government Graphic.pdf ● Getting Started with Instructions (1).pdf ● Three Branches Explained (1).pdf ● Name That Branch Blank.pdf ● Name That Branch (Key).pdf ● Who Got the Power? (Key).pdf ● Who Got the Power?.pdf
CS objectives addressed:	Students will be able to: ● Program a robot mouse to identify who has the power in three branches of government ● Test and debug a program to achieve a specific goal Standards: ● **AP.1B.8**—Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (Program Development) ● **AP.1B.9**—Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. ● **AP.1B.10**—Describe choices made during program development using code comments, presentations, and demonstrations.
Time needed:	**Total Time: 60 minutes** (Each day of this 3-day lesson will include Engagement 10 min; Explore 35 min; and Explain 15 min.) ● Engagement **10 min** ● Explore **35 min** ● Explain **15 min**
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Robot (Some examples include: Code and Go Mouse, Botley, Dash) ● Robot Game ● Three Branches of Government Graphic ● Three Branches of Government Explained (Teacher Resource) ● Name the Three Branches

	● Name the Three Branches Key (Teacher Resource) ● Who Got the Power? Handout ● Who Got the Power? Handout Key (Teacher Resource) ● Getting Started with Instructions
Subject integrated:	Social Studies
Other standards addressed:	**H.3.2**—Explain the role of representative democracy in framing the American government.
Vocabulary:	Debugger: The process of removing existing errors Programmer : The individual who sets the instructions on a robot
Notes:	This lesson may take longer because you may need to teach them about the 3 branches of government prior to the lesson. Students will understand the worksheets better.