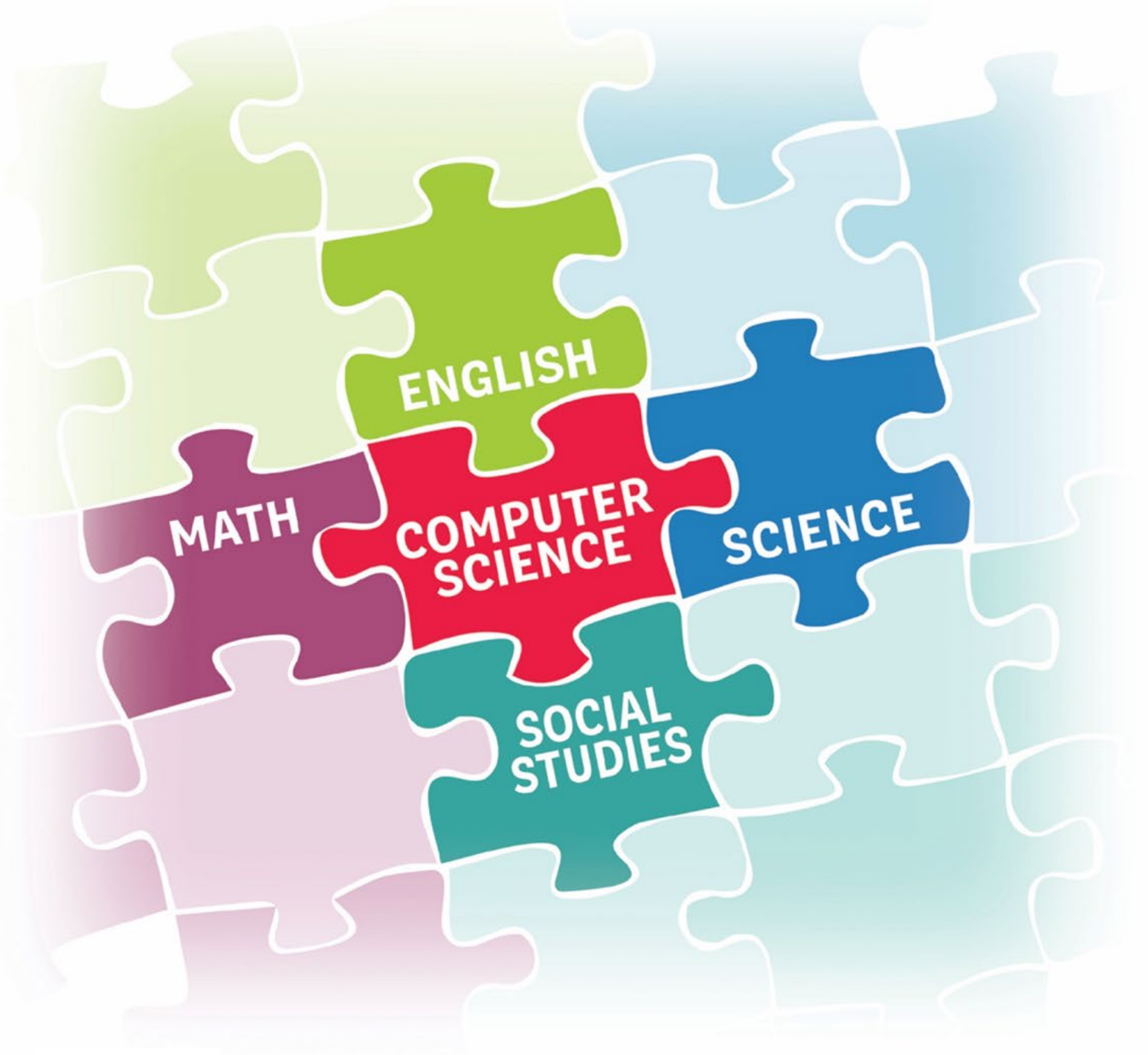


2023

Elementary Integration Guide

SIXTH GRADE

MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

Acknowledgements

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

Development Team

Melissa Banks, Director of Digital Learning, Mississippi Department of Education
Lizzie Brandon, Teacher, Houston High School
Lerenda Dixon, Teacher, McComb School District
Carrie Doubleday, Teacher, South Panola School District
Mary Dunaway, Teacher, Rankin County School District
Dr. Anthony Golding, Administrator, Tupelo Public School District
Shelly Hollis, Director, Center for Cyber Education
Courtney Hopper, Teacher, South Panola School District
Laura Sheffield, Instructional Coach, Tupelo Public School District
Amanda Taylor, Project Manager, Center for Cyber Education
Ann Thomas, Teacher, Lafayette County School District
Shelley Thompson, Project Manager, Center for Cyber Education
Dr. Louella Mack-Webster, STEM Program Supervisor, Mississippi Department of Education
Dr. Haley Wilson, Curriculum/Technology Specialist, Lafayette County School District

Funding for the development of this guide was provided by:

MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

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 were released on the CS4MS.org website for the 2022-2023 school year.

A team was constructed to look at the needs for having a sixth-grade integration guide that would benefit each school and/or district while addressing the plethora of options for scheduling sixth grade students. This integration guide contains a breakdown of content by integrated subjects only. The guide is divided into six subject areas: English Language Arts, Mathematics, Science, Social Studies, Social and Emotional Learning, and Digital Citizenship. In addition to a lesson overview and links to required resources, each lesson plan maps to a Mississippi Computer Science Standard and another 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. 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.

Table of Contents by Integrated Subjects

30 – 31	Pizza Party Data	Digital Literacy Unplugged	DA.2.2	6.SP.5
32 – 33	Robotic Shape Trace	Robotics Coding	Ap.2.3a, AP.2.4a, AP.2.8a	6.G.1
34 – 35	Number Line Animation	Robotics Coding	Ap.2.3a, Ap.2.4a, AP.2.8a	6.NS.9c, 6.NS.9d
36 – 37	Using Algorithms to Find Answers	Coding Unplugged	Ap.2.1, AP.2.1a	6. EE.6

Science

Page(s)	Lesson Title	Topic(s)	CS Standard	Core Subject Standard
39 – 40	Living vs. Non-Living	Coding	CS.2.2a, AP.2.7a, AP.2.7b, AP.2.10a, AP.2.4a, AP.2.5a, AP.2.8a, AP.2.9a, AP.2.9b, AP.2.9c	L.6.1.1
41 – 42	Organelle Robotics	Robotics Coding	AP.2.4a, AP.5.a, AP.2.8a, AP.2.9a, AP.2.9b, AP.2.9c, AP.2.10b	L.6.1.3
43 – 44	Scratch Relationships	Coding	AP.2.4a, AP.2.5a, AP.2.6a, AP.2.9a, AP.2.9b, AP.2.9c	L.6.3.4
45 – 46	Energy In an Ecosystem: Part 1	Digital Citizenship	AP.2.4a, AP.2.5a, AP.2.6a, AP.2.9a, AP.2.9b, AP.2.9c	L.6.3.5
47 – 48	Energy In an Ecosystem: Part 2	Coding	AP.2.4a, AP.2.5a, AP.2.6a, AP.2.9a, AP.2.9b, AP.2.9c	L.6.3.5
49 – 50	Dichotomous Key Robotics	Robotics Coding	AP.2.4a, AP.2.5a, AP.2.8a, AP.2.10b	L.6.4.2
51 – 52	Coding Kingdoms	Coding	AP.2.4a, AP.2.5a	L.6.4.2

53 – 54	Scratching the Surface of Newton's Law	Coding	AP.2.4a, AP.2.5a	P.6.6.1
55 – 56	Energy and Motion	Coding	AP.2.4a, AP.2.5a	P.6.6.7
57 – 58	Solar System	Coding	AP.2.4a, AP.2.5a, AP.2.9a, AP.2.9b, AP.2.9c	E.6.8.4
59 – 60	Moon Phases	Robotics Coding	AP.2.4a, AP.2.5a	E.6.8.6
61 – 62	Cell Explorations	Coding	AP.2.3a, AP.2.8a, AP.2.10a	L.6.1.3

Social Studies

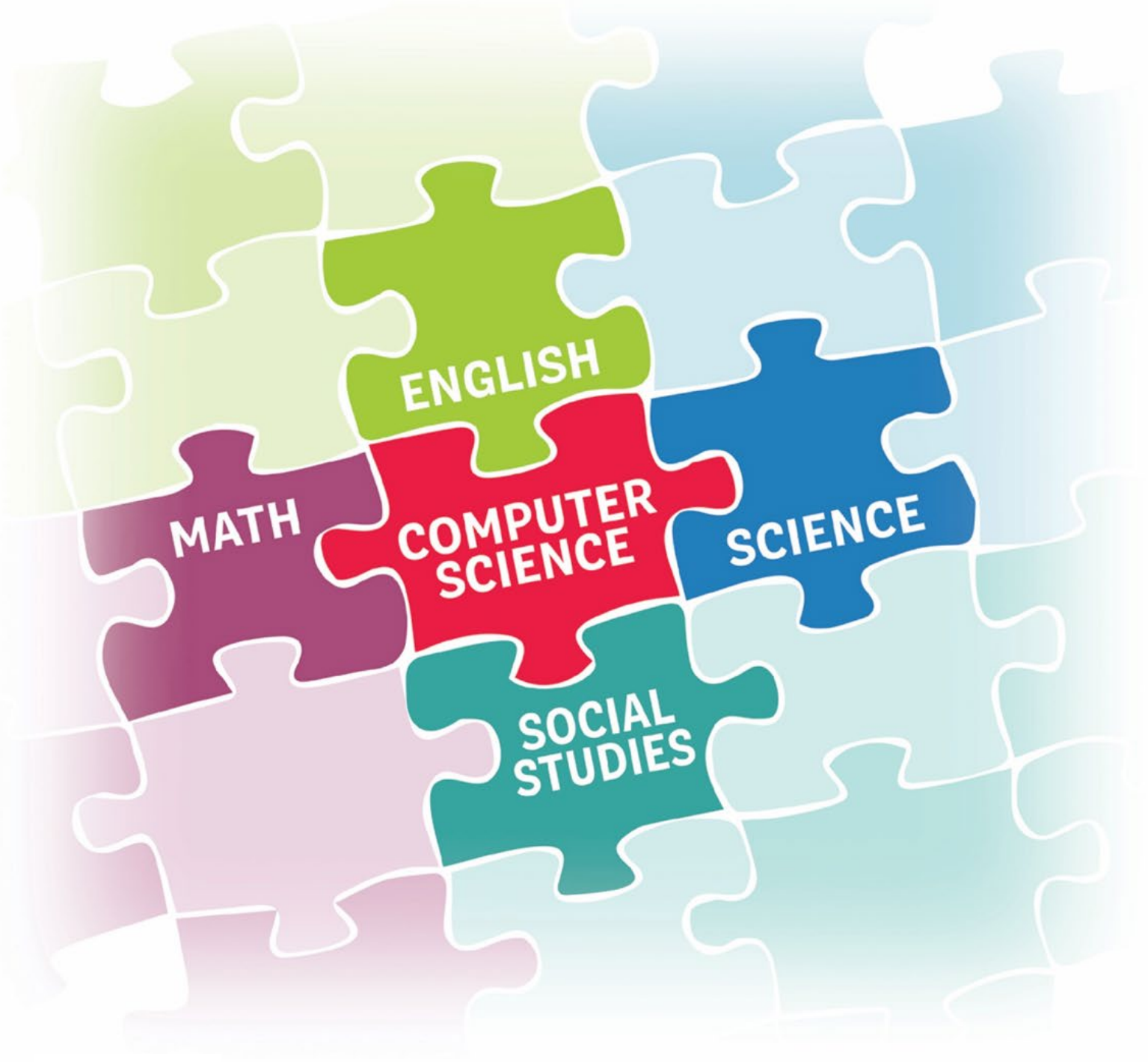
Page(s)	Lesson Title	Topic(s)	CS Standard	Core Subject Standard
64 – 65	Native American Star Quilts – Code.org Star Quilts Module: Lesson 1	Unplugged	AP.2.3a, AP.2.4a	H.6.1, H.6.1.1, H.6.1.2, H.6.1.3, 6.G.1
66 – 67	Getting Started with the Artist – Code.org Star Quilts Module: Lesson 2	Coding	Ap.2.3a, AP.2.4a	H.6.1, H.6.1.1, H.6.1.2, H.6.1.3, 6.G.1
68 – 69	Code Your Star Quilt – Code.org Star Quilts Module: Lesson 3	Coding	Ap.2.3a, AP.2.4a	H.6.1, H.6.1.1, H.6.1.2, H.6.1.3, 6.G.1
70	Designing for Accessibility	Unplugged	CS.2.1, CS.2.1a	G.6.7, G.6.7.3
71 – 72	Finding Credible News: How do we find credible information on the internet?	Digital Literacy	IC.2.4, IC.2.1a	CI.6.2, CI.6.2.1
73 – 74	Cybersecurity – Simple Encryption	Coding	NI.2.3a	H.6.1.1

Social and Emotional Learning

Digital Citizenship

Appendices

Resources	
Computing resources	• Code.org • Common Sense Digital Media • Scratch • Hour of Code • CS First - Google • Codesters • Tynker • VEX VR
CS4MS website materials	• 2018 Mississippi Computer Science Standards • CS4MS Website
Teacher/student accounts	• Code.org • Common Sense Digital Media • Scratch
For help with this guide	• Contact Mississippi State University's Center for Cyber Education: www.tinyurl.com/ccehelpdesk



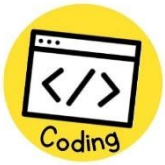
2023

Elementary Integration Guide

LANGUAGE ARTS

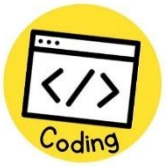
MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

Code Words

Lesson overview:  	**Purpose:** ● Students will learn about coding history and terminology. ● Students will learn to code by creating an animated story online. **Lesson:** Warm Up: ● Follow prompts on the Code Words - Coding and ELA Activities - Lesson Plan ● Students will brainstorm the different ways 'code' is used (zip code, bar code, code of honor, dress code, cracking the code) ● Hand out Scholastic - Student magazine (Print or direct students to navigate to the magazine online). Have students read independently or with a partner. Challenge them to use context to better understand the bolded words as they read and direct them to complete the quiz on the back page. Quiz answer key: 1. C; 2. B; 3. D; 4. A; 5. D; 6. B. ● Review the central ideas in the magazine as a class. Prompt students to identify vocabulary words in the magazine that were new to them. Go over the meanings of the words as a class. (Extension: Have students find out five more facts about one of the people in the "Great Moments in Coding History" sidebar.). ● Explain that computer coding is about solving problems by breaking a project into smaller tasks. Prompt students to name other things they do that depend on the same kind of logic and problem-solving involved in coding (examples: following a recipe to bake a cake, learning to play a musical instrument, making a craft project). Collect a list on the board. Activity: ● Follow prompts on the Adventure on the High Seas lesson plan ● Try the High Seas Introduction - CS First activity on the CS First site. Students will pick two "sprites," or characters, place them in a boat, and create dialogue to tell a story. ● Prompt students to imagine a story inspired by a historical figure, current event, or fictional character they've studied in class. Is one of the characters an early explorer to the Americas? A scientist studying climate change? A fictional character (or two) from their favorite novel? Encourage them to get creative! ● Tell students to spend a few minutes organizing their story plot on paper or in a digital document. Prompts: Are your characters traveling somewhere exciting? Are they searching the ocean for something? Are they lost? ● Remind students to use realistic sounding dialogue. Based on their knowledge of the characters' historical background, interests, motivations, and personality, what might the conversation be about? What type of "voice" would each character have? How does their dialogue help move the story line forward? Wrap Up: ● Have students share their animations with their classmates.
Lesson links/resources:	● Code Words - Coding and ELA Activities - Lesson Plan ● Scholastic - Student magazine ● Adventure on the High Seas overview - CS First ● Adventure on the High Seas lesson plan

	• High Seas Introduction - CS First
CS standards addressed:	**AP.2.2b** Students should use naming conventions to improve program readability. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.8a** Students will test programs by considering potential errors, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive numbers). **AP.2.10** Document programs to make them easier to follow, test, and debug.
Time needed:	**Total time: 60 Mins** • Warm Up: 20 Mins • Activity: 30 Mins • Wrap Up: 10 Mins
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RL.6.1** Cite textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text. **RL.6.4** Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings. **SL.6.1c** Pose and respond to specific questions with elaboration and detail by making comments that contribute to the topic, text, or issue under discussion.
Vocabulary:	**Algorithm** – the steps you take to reach a goal or solve a problem. **Sequence** – doing things in a specific order. **Loop** – repeating a sequence. **Program** – using code to make a computer or other device perform a certain action. **Event** – an action that causes the program to begin or respond.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at CS First

Illustrating with Coding

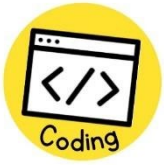
Lesson overview:  	**Purpose:** ● The students will create coded animations for various scenes to illustrate a certain part of a story. Once each student, or group of students, complete their project, the entire story should be represented through the animations. **Lesson:** Warm Up: ● Follow prompts on the Code.org - Lesson 10 Mini Project - Captioned Scenes Lesson Plan ● Students will read a designated section of a story, provided by the teacher. Activity: ● Students will plan out (sketch) their scene on paper. ● Students will create a background to represent their part of the story on Code.org - Lesson 10 Mini-Project - Captioned Scenes Code Studio. ● Students will add Sprite(s) and animate them to represent their part of the story. ● Students will add a text to their animation. Wrap Up: ● Groups will share their scenes to the class to retell the entire story. The students can present their projects to the class, share their projects through Scratch, and/or post the link to a class learning management system.
Lesson links/resources:	● Code.org - Lesson 10 Mini Project - Captioned Scenes Lesson Plan ● Code.org - Lesson 10 Mini-Project - Captioned Scenes
CS standards addressed:	**AP.2.2b** Students should use naming conventions to improve program readability. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.8a** Students will test programs by considering potential errors, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive numbers). **AP.2.10** Document programs in order to make them easier to follow, test, and debug.
Time needed:	**Total time: 60 Mins** ● Warm Up: 10 Mins ● Activity: 35 Mins ● Wrap Up: 15 Mins
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	ELA

Other standards addressed:	**RL.6.3** Describe how the plot of a literary text unfolds in a series of episodes as well as how the characters respond or change as the plot moves toward a resolution.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org
Design the Device	

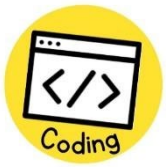
Lesson overview: 	**Purpose:** ● Students will compare different forms of technology (laptops, phones, tablets) on the Open Classrooms - Design for the Device. **Lesson:** Warm Up: ● Assign students to work in pairs to analyze which features of a device of their choice that could be improved. Activity: ● Student pairs will sketch out the design of their improved device and create a 1–2-minute advertisement or commercial that lists the benefits of the improved design. Wrap Up: ● Pairs will present their designs and advertisements to the class.
Lesson links/resources:	● Open Classrooms - Design for the Device
CS standards addressed:	**CS.2.1** Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices. **CS 2.1a** Students should make recommendations for existing devices (e.g. a laptop, phone, or tablet) or design their own components or interface (e.g. create their own controllers)
Time needed:	**Total time: 60 Mins** ● Warm Up: 10 Mins ● Activity: 30 Mins ● Wrap Up: 20 Mins
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.1** Cite textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text **RI.6.7** Integrate information presented in different media or formats as well as in words to develop a coherent understanding of a topic or issue **W.6.2** Write informative explanatory texts to examine a topic and convey ideas, concepts, and information through the selection, organization, and analysis of relevant content **W.6.2d** Use precise language and domain-specific vocabulary to inform about of explain the topic **W.6.9** Draw evidence from literary or informational texts to support analysis, reflection, and research
Vocabulary:	

Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Open Classrooms (when entering your DOB use DD/MM/YYYY).
Weather Design	
Lesson overview:	**Purpose:** ● Students will learn how to collect data, interpret information presented in different media formats, and write an informational conclusion about how the weather various in different parts of the world. **Lesson:**

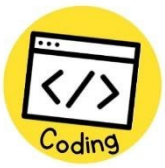
	Warm Up: • Students will collect weather data from a weather app or website for a week from specific areas of the globe. Activity: • Students will enter data into a spreadsheet (Microsoft Excel or Google Sheets) and create graphs to model the data. Wrap Up: • Students will then write reports to compare the data from the different areas to make inferences on how weather changes depending on time of year and location
Lesson links/resources:	• Weather Bug • Google Sheets
CS standards addressed:	**CS.2.2** Design projects that combine hardware and software components to collect and exchange data **CS 2.2a** Students will design projects that use both hardware and software to collect and exchange data
Time needed:	**Total time: 60 Mins** • Warm Up: (Students research and record weather information for their location each day of the week) • Activity: 40 Mins • Wrap Up: 20 Mins
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.7** Integrate information presented in different media or formats (e.g., visually quantitatively) as well as in words to develop a coherent understanding of a topic or issue **W.6.2** Write informative/explanatory texts to examine a topic and convey ideas, concepts, and information through the selection, organization, and analysis of relevant content. **W.6.2d** Use precise language and domain-specific vocabulary to inform about or explain the topic **W.6.2f** Provide a concluding statement or section that follows from the information or explanation presented. **W.6.6** Use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills. **W. 6.7** Conduct short research projects to answer a question, drawing on several sources and refocusing the inquiry when appropriate.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Open Classrooms This activity can be completed in Excel instead of Google Sheets.

Scratch Story Board	
Lesson overview:  	**Purpose:** ● Students will apply what they have learned about a narrative story and how a particular sentence, chapter, or scene fits into the overall structure of the text and contributes to the development of the theme, setting, or plot. **Lesson:** Warm Up: ● Student(s) will create a 2–3-minute story as the narrator. Activity: ● Student(s) will create their narrative story in Scratch. ● Students will create a background in Scratch that is appropriate for their story. ● Students will add a minimum of two Sprites and eight sets of text to their storyboard in Scratch. Wrap Up: ● Students will justify how the background (setting) of their story supports the story. ● Students will share their stories with their peers.
Lesson	● Scratch

links/resources:	
CS standards addressed:	**AP.2.3** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. **AP.2.3a** Students will design and develop programs that combine control structures
Time needed:	**Total time: 60 Mins** ● Warm Up: 15 Mins ● Activity: 35 Mins ● Wrap Up: 10 Mins
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RL.6.5** Analyze how a particular sentence, chapter, scene, or stanza fits into the overall structure of a text and contributes to the development of the theme, setting or plot. **RL.6.7** Compare and contrast the experience of reading a story, drama, or poem to listening to or viewing an audio, video, or live version of the text, including contrasting what they "see" and "hear" when reading the text to what they perceive when they listen or watch. **W.6.6** Use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

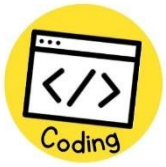
The World of Ozaria	
Lesson overview:  	**Purpose:** ● Enter the world of Ozaria where you become a hero in an epic adventure. You must use the power of coding to defeat a darkness that has taken over the world! Along the way, you'll meet interesting characters and travel to different lands, practicing coding concepts like sequences, loops, debugging, and decomposition. In the end, you'll design a playable game that you can share with your friends! **Lesson:** Warm Up: ● Follow prompts on the Oz Chp. 1 - Lesson Slides (Note: slides 1 through 7 are instructions for how to use the Lesson Slides) Activity: ● Students will play through the suggested levels on the "Independent Practice Slide Wrap Up: ● Use the check-in section as an opportunity for students to reflect on what they have learned in the lesson.
Lesson links/resources:	● Oz Chp. 1 - Lesson Slides ● Ozaria - Hour of Code
CS standards addressed:	**CS.2.3** Systematically identify and fix problems with computing devices and their components. **AP 2.3** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. **AP.2.7** Incorporate existing code, media, and libraries into original programs

	and give attribution.
Time needed:	**<u>Total time:</u> 60 Mins** ● Warm Up: 15 Mins ● Activity: 35 Mins ● Wrap Up: 10 Mins
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.2**: Determine a central idea of a text and how it is conveyed through particular details; provide a summary of the text distinct from personal opinions or judgments. **RI.6.7**: Integrate information presented in different media or formats (e.g., visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue. **RI.6.10**: By the end of the year, read and comprehend literary nonfiction in the grades 6-8 text complexity band proficiently, with scaffolding as needed at the high end of the range. **W.6.6**: Use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills, **W.6.7**: Conduct short research projects to answer a question, drawing on several sources and refocusing the inquiry when appropriate.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at https://www.ozaria.com/

Coding in Python	
Lesson overview:  	**Purpose:** ● In this lesson, students will focus on how events can be combined to create an interactive game. This lesson involves multiple events at once, some controlling the player sprite, another controlling the game environment, and another that awards the player points for doing the right action. Students will be able to use shapes, sprites, variables, and events to create an object-drop game. **Lesson:** (Review the lesson before presenting it to students.) Follow the steps outlined in the Codesters Feed the Fish - guide. Work as a whole group, small group, or independently and at a pace that is fitting for your group of students.
Lesson links/resources:	● Codesters Feed the Fish - guide ● Feed the Fish - Hour of Code
CS standards addressed:	**AP.2.2** Create clearly named variables that represent different data types and perform operations on their values. **AP.2.4** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs **AP.2.5** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs
Time needed:	**Total time: 60 Mins** (This lesson can be started and then students complete the various stages when time allows.)
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.7** Integrate information presented in different media or formats (e.g., visually quantitatively) as well as in words to develop a coherent understanding of a topic or issue **W.6.6** Use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient

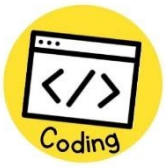
	command of keyboarding skills.
Vocabulary:	**sprite:** a character, shape, or text object that we add to our scene or game. **event:** a block of code that has a specific task and must receive a signal to run. **interval event:** a block of code that runs once every time a specified number of seconds has passed.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

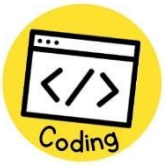
Recycle and Reuse: UN Sustainable Development Goals Project

Lesson overview: 	**Purpose:** Students will apply coding concepts to create a Responsible Consumption and Production project and read about the Sustainable Development Goals. **Lesson:** Warm Up: ● "17 Sustainable Development Goals" video ● Ask students to read about *Goal 12: Responsible Consumption and Production* Activity: ● Have students complete the "Responsible Consumption and Production" activity sheet. ● Students will create a meaningful coding project about responsible consumption and production. Wrap Up: ● Have students share their projects with the class, or classmates.
Lesson links/resources:	● Responsible Consumption and Production JavaScript Teacher Guide ● Responsible Consumption (JavaScript) Project - Hour of Code
CS standards addressed:	**AP.2.4.** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. **AP. 2.8** Systematically test and refine programs using a range of test cases.
Time needed:	**Total time: 60 Mins** (If students do not finish the complete assignment in class, they can work on the project at home.)
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.4.** Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings. **RI.6.7.** Integrate information presented in different media or formats (e.g. visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue. **SL.6.1.** Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 6 topics, texts, and issues, building on others' ideas and expressing their own clearly.
Vocabulary:	**Code** - the language that tells a computer what to do. **Sequence** -the order in which steps or events happen. **Function** - a set of known actions that the computer can perform. **Variable** - stores a value, such as a number or a string of text, at a named location. **Argument** - value passed into a function.

	Parameter - an extra piece of information that is passed into a function. **Loop** - an action that repeats one or more commands over and over
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at tynker.com

Tree Planting: UN Sustainable Development Goals Project

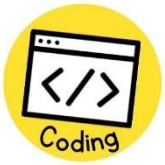
Lesson overview: 	**Purpose:** Part of the UN Sustainable Development Goals project, this Python project will have you create your own tree-planting game. **Lesson:** Warm Up: ● "17 Sustainable Development Goals" video ● Ask students to read about *Goal 15: Life on Land* Activity: ● Have students complete the "Life on Land" activity sheet. ● Students will create a meaningful coding project about life on land. Wrap Up: ● Have students share their projects with the class, or classmates.
Lesson links/resources:	● Life on Land Python Teacher Guide ● Life on Land (Python) Project - Hour of Code
CS standards addressed:	**AP.2.4.** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. **AP.2.8.** Systematically test and refine programs using a range of test cases.
Time needed:	**Total time: 60 Mins** If students do not finish the complete assignment in class, they can work on the project at home.
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.4**. Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings. **RI.6.7.** Integrate information presented in different media or formats (e.g. visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue. **SL.6.1.** Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 6 topics, texts, and issues, building on others' ideas and expressing their own clearly.
Vocabulary:	**Code** - the language that tells a computer what to do. **Sequence** -the order in which steps or events happen. **Function** - a set of known actions that the computer can perform. **Variable** - stores a value, such as a number or a string of text, at a named location. **Argument** - value passed into a function. **Parameter** - an extra piece of information that is passed into a function. **Loop** - an action that repeats one or more commands over and over
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at tynker.com

Time Travel	
Lesson overview:  	Learn basic coding concepts to correct mysterious mishaps throughout history! Travel back in time to save the future in this free Hour of Code lesson in Minecraft: Education Edition. Players will choose their own adventure and connect with great innovators and inventions in science, architecture, music, engineering, and more. Follow the steps below to get started! • This lesson requires a free download of the educational version of Minecraft. Step 1 in the Minecraft Hour of Code 2021 link. • Teachers who are unfamiliar with Minecraft should spend time familiarizing themselves with the game and how the lesson works. Pay special attention to the Teacher Prep and Notes section in the Hour of Code: TimeCraft - Minecraft Education guide.
Lesson links/resources:	• Hour of Code: TimeCraft - Minecraft Education guide • Minecraft Hour of Code 2021

CS standards addressed:	**AP.2.1**. Use flowcharts and/or pseudocode to address complex problems as algorithms. **AP.2.3**. Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. **AP.2.4.** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
Time needed:	**Total time: 60 Mins** If students do not finish the complete assignment in class, they can work on the project at home.
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.4** Determine or clarify the meaning of unknown and multiple meaning words and phrases based on grade 6 reading and content, choosing flexibly from a range of strategies **RI.6.7.** Integrate information presented in different media or formats (e.g., visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue.
Vocabulary:	**Computer Science** – the study of using the power of computers to solve problems.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org


ENGLISH
MATH
COMPUTER SCIENCE
SCIENCE
SOCIAL STUDIES

2023

	Data Visualization	
	Lesson overview:  	**Purpose:** ● This lesson builds off the concepts for bar charts and histograms that have already been introduced to students. Students will practice making conclusions from charts and learn to use the Data Visualizer in App Lab to create two different kinds of charts: a bar chart, and a histogram. This tool is designed to quickly connect students with real-world datasets and make it easy to create visualizations from data without learning how to navigate a more complex tool. They will also have access to several real-world datasets that they can use to create their charts. **Lesson:** Warm Up: This lesson is intended to enrich or reinforce existing lessons on bar charts and histograms. It is not meant to introduce new core content to students. Students should come into this lesson with the following prior knowledge and skills: ● Review how to read a bar chart, understanding what the values on the x and y axes represent. ● Know how to read a histogram, understanding that the range of values represent. Activity: ● The lesson begins with a quick prompt to review the reasons charts are useful for looking at data. ● Students then practice creating a bar chart in the Data Visualizer in App Lab. Then they learn how to make histograms for building charts in instances where bar charts may not be useful. ● Students then have a chance to explore different real-world datasets and see how making charts can help communicate information about a topic they care about. Wrap Up: ● Discuss how to determine when to use a bar chart and a histogram. ● Discuss the consequences of using data that is not accurate.
	Lesson links/resources:	Code.org - Lesson 1: Data Visualization
	CS standards addressed:	**DA.2.3a** Students will refine computational models by considering which data points are relevant, how data points relate to each other, and if the data is accurate.
	Time needed:	**Total time: 45 Mins** ● Warm Up: 10 Mins ● Activity: 35 Mins ● Wrap Up: 15 Mins
	Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access

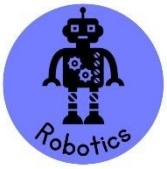
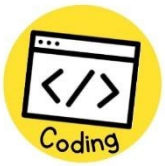
	Subject integrated:	Math
	Other standards addressed:	**6.SP.4** Display numerical data in plots on a number line, including dot plots, histograms, and box plots. **6.SP.5** Summarize numerical data sets in relation to their context, such as by: a. Reporting the number of observations. b. Describing the nature of the attribute under investigation, including how it was measured and its units of measurement. c. Giving quantitative measures of center (median and/or mean) and variability (interquartile range), as well as describing any overall pattern and any striking deviations from the overall pattern with reference to the context in which the data were gathered. d. Relating the choice of measures of center and variability to the shape of the data distribution and the context in which the data were gathered.
	Vocabulary:	
	Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Pizza Party Data

Lesson overview:  	**Purpose:** ● Create a bar chart based on a set of data. ● Explain why a set of data must be cleaned before a computer can use it. ● Identify and remove irrelevant data from a data set. **Lesson:** Warm up: ● The teacher will have students analyze the data displayed in the three representations from a survey.

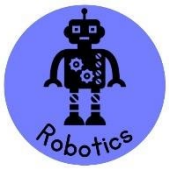
	• Journal prompt - Which one of these makes it easiest for a human to make a decision about which pet is the most popular? Which one makes it easiest for a computer to make a decision? • Go over vocabulary. Activity: • Students will look at the Structuring Data 2021 Activity Guide using it to create a bar chart for the raw data and have them answer the reflection questions at the bottom of the guide. • Students will discuss with a partner their answers to the reflect questions then discuss them as a class. • Teacher will demonstrate the Pizza Party Data App with the class. • Students will use the Pizza Party Data App to clean up the data from their interpretations so that the computer can organize the information properly. Wrap up: • Students will answer the following questions in their journals: Can you think of a time in the past when you had data collected about you, maybe by filling out a form? What do you think were some strategies this form used to help make sure it collected clean data?
Lesson links/resources:	• Code.org Lesson 10: Structuring Data • Structuring Data 2021 Activity Guide • Pizza Party Data App
CS standards addressed:	**DA.2.2** Collect data using computational tools and transform the data to make it more useful and reliable.
Time needed:	**Total time: 45 Mins** • Warm Up: 5 Mins • Activity: 35 Mins • Wrap Up: 5 Mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Structuring data slides (code.org) • Code.org account Students: • Computer/tablet with internet access • Structuring Data 2021- Activity guide (code.org) • Code.org account
Subject integrated:	Math
Other standards addressed:	**6.SP.5** Summarize numerical data sets in relation to their context
Vocabulary:	**Raw data** - The way information is first collected
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

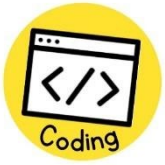
Robotic Shape Tracer

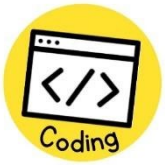
Lesson overview:  	**Purpose:** • Students will code a virtual robot to trace various shapes (i.e., triangles, quadrilaterals, and parallelograms). • Students will review the name/classification of shapes. **Lesson:** Warm up: • The teacher will review the characteristics of triangles and how to classify them by sides and angles. • The teacher will review the characteristics of quadrilaterals and parallelograms. Activity: This activity can be completed individually, or in small groups. • The student(s) will draw/select a shape from the VEXcode VR selection playground titled, "Shape Tracer." (This can be done individually or in small groups) • Students will list the characteristics and name of their shape on paper. • Students will code the virtual robot to trace the shape they selected. • Student(s) will calculate side lengths for their shape and find the area and perimeter. • Students who have the same shape, will record their time for the robot to trace their shape. The student(s) with the fastest robot (changing the velocity of the robot will affect the speed), will win the robot "competition." • Students will present their shape (characteristics, name, classification, etc.) to the class and demonstrate their robot tracing their shape. Wrap up: • Reflection and Sharing: Questions to discuss: what code command controls the speed of the robot? What angle degrees were needed to code the robot to trace triangles? Quadrilaterals? Why are those angle measurements the same or why are they different? Would there by a way of simplifying the total number of code lines – what is that called?
Lesson links/resources:	• VEX.code VR Activity_Tracing Unique Shapes • VEXcode VR (choose "select playground" in the top right corner, and select "Shape Tracer" activity)
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.8a** Students will test programs by considering potential errors, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive numbers).
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **30 Mins** • Wrap Up: **20 Mins**

Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound Students: ● Computer/tablet with internet access ● VEX VR Coding Playground
Subject integrated:	Math
Other standards addressed:	**6.G.1** Find the area of right triangles, other triangles, special quadrilaterals, and polygons by composing into rectangles or decomposing into triangles and other shapes; apply these techniques in the context of solving real-world and mathematical problems.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at vr.vex.com

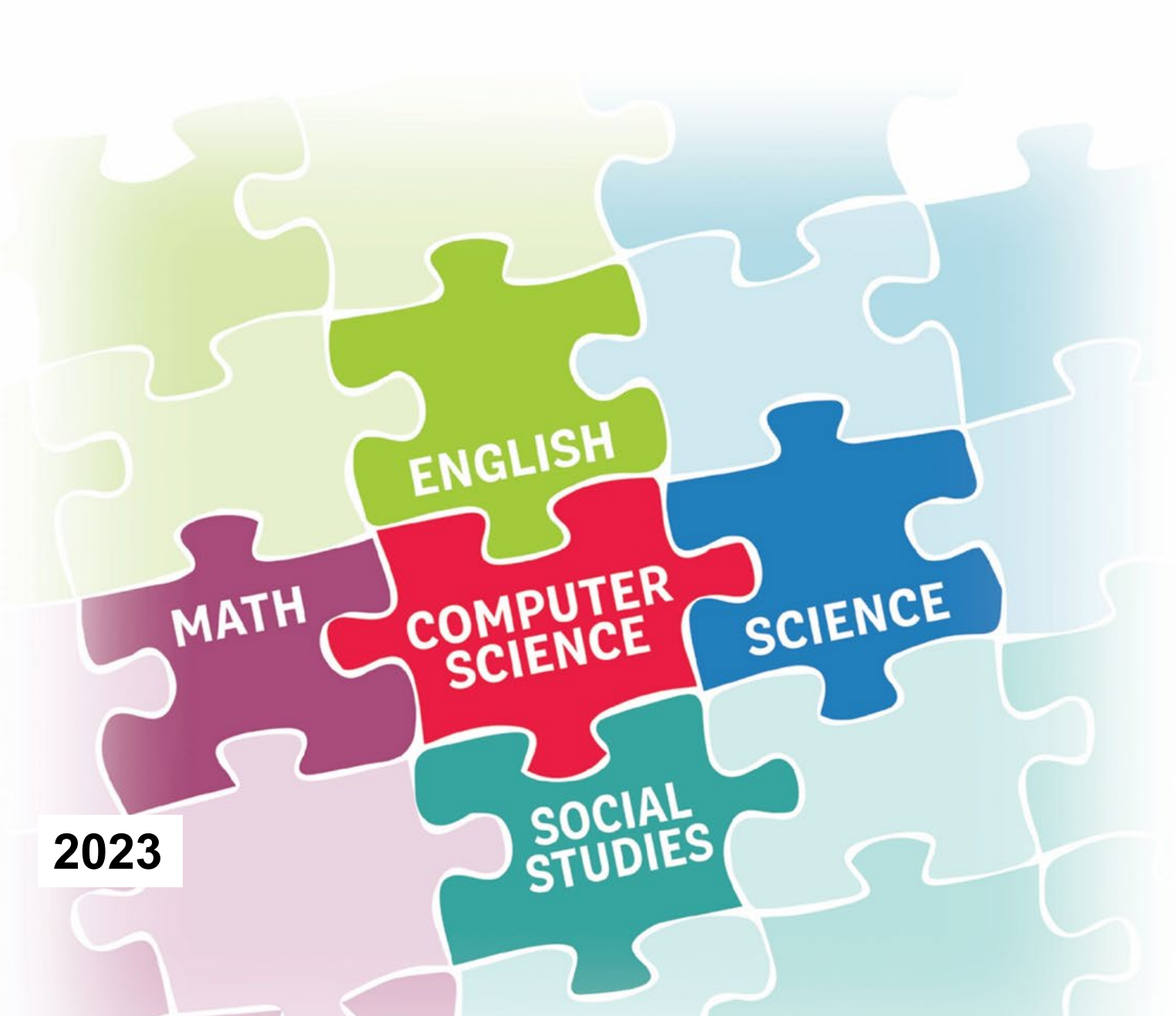
Number Line Animation

Lesson overview: 	**Purpose:** ● Students will create an animation of a robot or Sprite moving on a number line to represent addition and subtraction of integers. ● Students will create 3-5 addition or subtraction integer problems and code the robot or Sprite to move along the number line to represent the problem and solution. **Lesson using Scratch:** This activity can be completed individually, or small groups. Warm up: ● The teacher will review the rules for adding and subtracting integers.

	Activity: ● The teacher will facilitate how to insert the number line graphic into the Scratch stage. ● The teacher can provide 3-5 adding or subtraction of integers problems or the students can create their own. ● The student(s) will create and animate a Sprite to move along the number line to demonstrate the math problems. Wrap up: ● The student(s) will share at least one animation of their math problems. **Lesson using a robot:** This activity can be completed individually, or small groups. Warm up: ● The teacher will review the rules for adding and subtracting integers. Activity: ● The teacher will provide a number line drawn on butcher paper or the floor. (The students can make their own number line if time allows) ● The teacher will provide 3-5 adding or subtraction of integers problems. ● The student(s) will program a codeable robot to move along the number line to demonstrate each math problem. The students will write their program next to each math problem provided. ● The students will create 1-2 math problems and share with the group next to them for programming and solving. Wrap up: ● The student(s) will share at least one animation of their math problems.
Lesson links/resources:	● Adobe Stock Number Line ● Scratch
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.8a** Students will test programs by considering potential errors, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive numbers).
Time needed:	**Total time: 60 Mins** ● Warm Up: **10 Mins** ● Activity: **40 Mins** ● Wrap Up: **10 Mins**
Materials needed:	Teacher: ● Codable robot (e.g., Code and Go Mouse, Dash Robot, Botley) ● Computer ● Projector/smartboard with sound Students: ● Computer/tablet with internet access
Subject integrated:	Math

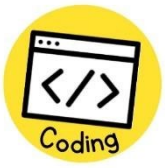
Other standards addressed:	**6.NS.9c** Understand subtraction of integers as adding the additive inverse, p – q = p + (–q). Show that the distance between two integers on the number line is the absolute value of their difference and apply this principle in real-world contexts. **6.NS.9d** Apply properties of operations as strategies to add and subtract integers.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch
Using Algorithms to Find Answers	
Lesson overview:   	**Purpose:** • Students will read about how programmers use bits, bytes, and algorithms to design programs to solve problems. • Students will use an algorithm to build a paper airplane that flies the longest distance. **Lesson:** Getting Started:: • Follow the prompts on the Real-Life Algorithms: Paper Airplanes Teacher Guide • Discuss what students already know about algorithms? • Watch Computer Science Basics: Algorithms video • Review vocabulary • What We Do Daily Activity: • Read the NewsELA article The Math Behind Bits and Bytes and answer quiz questions. • Create an algorithm to build a paper airplane using the Real-Life Algorithms: Paper Airplanes Worksheet • Use the algorithm to create a paper airplane Wrap Up: • Flash Chat
Lesson links/resources:	• Real-Life Algorithms: Paper Airplanes Teacher Guide

	• Real-Life Algorithms: Paper Airplanes Worksheet • Computer Science Basics: Algorithms video • The Math Behind Bits and Bytes
CS standards addressed:	**AP.2.1** Use flowcharts and/or pseudocode to address complex problems as algorithms. **AP.2.1a** Students will use pseudocode and/or flowcharts to organize and sequence an algorithm that addresses a complex problem, even though they may not actually program the solutions.
Time needed:	**Total time: 60 Mins** • Getting Started: **15 Mins** • Activity: **30 Mins** • Wrap Up: **5 Mins**
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access • Paper
Subject integrated:	MATH
Other standards addressed:	**6.EE.6** Use variables to represent numbers and write expressions when solving a real-world or mathematical problem; understand that a variable can represent an unknown number, or depending on the purpose at hand, any number in a specified set.
Vocabulary:	**Algorithm** – the steps you take to reach a goal or solve a problem. **Bit** - the smallest piece of information in a computer. **Byte** - a unit of digital information that consists of 8 bits.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at newsela Teachers may want to change the lexile score to make the article more appropriate for students.



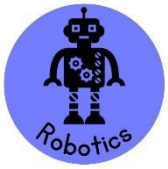
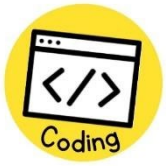
2023

Living vs. Non-Living

Lesson overview:  	**Purpose:** ● Living vs. Non-Living Clicker Game: Students will create an interactive clicker game that will help students learn about living and non-living things. ● The purpose in the game will be for the students to create a clicker game where they will differentiate between living (biotic) and non-living (abiotic) things. {Students should include examples such as viruses and bacteria.} **Lesson:** Warm-Up: ● As a class, watch the Scratch tutorial video to learn how to program a clicker game using Scratch. Activity: ● Students will work in groups of 2 – 4 to create a clicker game, using Scratch, that will earn point by selecting the correct answer and deducting points based on incorrect answers. ● Students must utilize various sprites, have at least one sound, have at least on background, and must assign point values to correct/incorrect answers. Wrap Up: ● Have students share their projects with other peers to try. ● Discuss the different ways students programmed their games. ● Discuss ways to help students debug their code if needed.
Lesson links/resources:	● How to Make a Clicker Game in Scratch Tutorial ● scratch.mit.edu
CS standards addressed:	**CS.2.2a** Students will design projects that use both hardware and software to collect and exchange data. **AP.2.7a** Students should use portions of code, algorithms, and/or digital media in their own programs and websites. **AP.2.7b** Students should test and refine programs using a range of test cases. **AP.2.10a** Students should provide documentation for end users that explains their artifacts and how they function. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.8a** Students will test programs by considering potential error, such as what will happen if a user enters invalid input (e.g. negative numbers and zero instead of positive number). **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions.
Time needed:	**Total time: 60 Mins** ● Warm Up: **10 Mins** ● Activity: **40 Mins** ● Wrap Up: **10 Mins** (This activity can be started in about 10 minutes. Students can complete

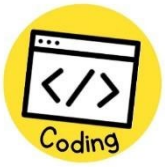
	their work as time allows.)
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound Students: ● Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.1.1** Use argument supported by evidence in order to distinguish between living and non-living things, including viruses and bacteria.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

Organelle Robotics

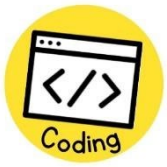
Lesson overview:   	**Purpose:** • Students will program a robot, or another student, to move around a grid to answer review questions for organelle review. **Lesson:** Warm-Up: • Whole group discussion on how to give commands, or program, a robot to move around a square grid. (Commands may include forward, backwards, turn left, turn right) Activity: • The teacher will create a grid 10 x 10 grid. This grid can be taped out on a table or the floor for a codable robot or student, or it can be completed using the Organelle Robotics Activity sheet. • There should be a clearly defined start square at the top left corner of the grid. • The teacher will print pictures or words of various organelles to place into the grid. (Example grid is provided in the lesson resources.) • The teacher will call out review questions, and the students will write a program for the "robot" to follow to get to the correct answer. Wrap-up: • Student should swap their program with a classmate or group to test. • Discuss why students' answers may vary (the code for the robot to get from start to the correct answer can be written in many different ways) Example: Teacher asks: Which organelle is considered the powerhouse of the cell? The student may answer: Forward, forward, turn right, forward, forward, forward, forward, forward (it would land on mitochondria).
Lesson links/resources:	• Organelle Robotics Activity • Printable Arrows
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.8a** Students will test programs by considering potential error, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive number). **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions. **AP.2.10b** Students should incorporate comments in their product (comments in the code).
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins**
Materials needed:	**Teacher:**

	• Robot Grid • Questions for Organelle Review **Students:** • Printable Arrows • Pencil and Paper • Robot/Student Robot
Subject integrated:	Science
Other standards addressed:	**L.6.1.3** Develop and use models to explain how specific cellular components (cell wall, cell membrane, nucleus, chloroplast, vacuole, and mitochondria) function together to support the life of prokaryotic and eukaryotic organisms to include plants, animals, fungi, protists, and bacteria (not to include biochemical function of cells or cell part).
Vocabulary:	
Notes:	

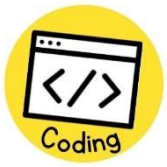
Scratch Relationships

Lesson overview:  	**Purpose:** • Students will work in groups of 2-4 to create a scratch program that will explain the following relationships: predation, competition, cooperation, and symbiotic relationships. **Lesson:** Warm-Up: • Review how to access and use Scratch. Show the Scratch Example Project to demonstrate what students will complete.

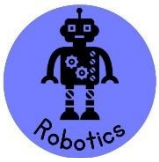
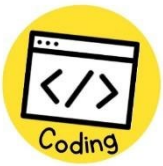
	• Review predation, competition, cooperation, and symbiotic relationships Activity: • Students will set the backdrop for at least one ecosystem. • Students will assign appropriate sprites to each relationship scenario. (Ex: Predator - Shark / Prey - Fish) {The sprites should be appropriate to the ecosystem. No sharks in the desert.} • Students will create a script for each scenario to the sprites by using the "say" option under the **Looks** tab. (Ex: Shark will say "I am the predator. I eat fish." Fish will say, "I am the prey. Sharks eat me for food. The shark will benefit from our relationship, where I do not benefit." Warm-up: • Once groups have completed their scenarios, teacher/students can conduct a gallery walk to view each groups project and provide positive constructive feedback.
Lesson links/resources:	• Scratch • CS First • Introduction to Interactive Presentation • Build Your Own Presentation • Scratch Example Project
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.6a** Students should begin to seek diverse perspectives throughout the design process to improve their computational artifacts. **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.3.4** Investigate organism interactions in a competitive or mutually beneficial relationships (predation, competition, cooperation, or symbiotic relationships).

Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch and csfirst.withgoogle.com
Energy in an Ecosystem: Part 1	
Lesson overview:  	**Purpose:** ● Students will work in groups to create a program that depicts energy flow through an ecosystem from producers to consumers to decomposers. **Lesson:** Warm Up: ● As a whole group, watch the Sprite Lab: Introducing Sprite Lab Video Activity: ● Students will log-on to code.org and open up a new Sprite Lab that they will rename to "Energy in an Ecosystem". ● Students will utilize the **world** tool to set an appropriate background representing the ecosystem. (There is a larger selection of backgrounds if you click "set background to" and click the down arrow. At the bottom of the background menu students will see "more". They can click that and have access to more backgrounds. ● Students will utilize the **sprites** tool to create the plants, animals, etc. (There is a larger selection of sprites if you click "Make new..." and click the down arrow. At the bottom of the sprite menu students will see "costumes" they will click that. Then they will click "New Costume"

	and select something from the costume's library, draw their own, or upload an image. ● Students will utilize tools such as events, behaviors, loops, variables, text, etc. to generate an informative sprite lab that explains energy flow through an ecosystem and each sprite's relevance. Wrap Up: ● Once students have completed their lab, have groups share their lab and provide/receive positive constructive feedback.
Lesson links/resources:	● Sprite Lab ● Sprite Lab - Sample Projects
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.6a** Students should begin to seek diverse perspectives throughout the design process to improve their computational artifacts. **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions.
Time needed:	**Total time: 60 Mins** ● Warm Up: **10 Mins** ● Activity: **40 Mins** ● Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.) This activity is a total of 120 minutes. The next lesson will be Energy in an Ecosystem: Part 2
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.3.5** Develop and use food chains, webs, and pyramids to analyze how energy is transferred through an ecosystem from producers (autotrophs) to consumers (heterotrophs, including humans) to decomposers.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

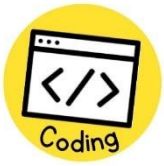
Energy in an Ecosystem: Part 2 (Continuation from "Energy in an Ecosystem: Part 1)	
Lesson overview:  	**Purpose:** ● Students will work in groups to create a program that depicts energy flow through an ecosystem from producers to consumers to decomposers. **Lesson:** Warm Up: ● As a whole group, watch the Sprite Lab: Introducing Sprite Lab Video Activity: ● Students will log-on to code.org and open up a new Sprite Lab that they will rename to "Energy in an Ecosystem". ● Students will utilize the **world** tool to set an appropriate background representing the ecosystem. (There is a larger selection of backgrounds if you click "set background to" and click the down arrow. At the bottom of the background menu students will see "more". They can click that and have access to more backgrounds. ● Students will utilize the **sprites** tool to create the plants, animals, etc. (There is a larger selection of sprites if you click "Make new..." and click the down arrow. At the bottom of the sprite menu students will see "costumes" they will click that. Then they will click "New Costume" and select something from the costume's library, draw their own, or upload an image. ● Students will utilize tools such as events, behaviors, loops, variables, text, etc. to generate an informative sprite lab that explains energy flow through an ecosystem and each sprite's relevance. Wrap Up: ● Once students have completed their lab, have groups share their lab and provide/receive positive constructive feedback.

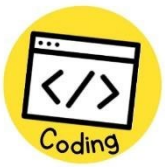
Lesson links/resources:	• Sprite Lab • Sprite Lab - Sample Projects
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.6a** Students should begin to seek diverse perspectives throughout the design process to improve their computational artifacts. **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.3.5** Develop and use food chains, webs, and pyramids to analyze how energy is transferred through an ecosystem from producers (autotrophs) to consumers (heterotrophs, including humans) to decomposers.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Dichotomous Key Robotics	
Lesson overview:   	**Purpose:** ● Students will use review various specimens' and their groups. ● Student will program a robot to move about the dichotomous key to group each specimen. **Lesson:** Warm-Up: ● Watch the tutorial video as a whole group and discuss the assignment expectations. (The dichotomous key and activity are based on the Amoeba Sisters video.) Activity: ● Students will determine which group each specimen belongs to by using the dichotomous key. ● Once the students have completed the dichotomous key, they will program their "robot" to: ➢ Travel through the specimen's name first, then the group that it belongs with. ➢ For each specimen, the student will need to begin at the specified start area. Wrap-Up: ● Students will need to record their programs into the chart provided. If time allows, have students share their projects.
Lesson links/resources:	● Dichotomous Key Robotics ● Dichotomous Keys: Identification Achievement Unlocked
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.8a** Students will test programs by considering potential error, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive number). **AP.2.10b** Students should incorporate comments in their product (comments in the code).
Time needed:	**Total time: 60 Mins** ● Warm Up 10 mins ● Activity 40 mins

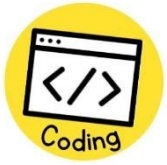
	● Wrap Up 10 mins
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.4.2** Use classification methods to explore the diversity of organisms in kingdoms (animals, plants, fungi, protists, bacteria). Support claims that organisms have shared structural and behavioral characteristics.
Vocabulary:	
Notes:	

Coding Kingdoms

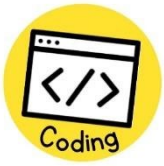
Lesson overview:  	**Purpose:** ● Students will apply their knowledge of kingdoms to develop a Scratch animation or clicker game. **Lesson:** Warm Up: ● Review kingdoms with the class as a whole group. ● Review the How to make a presentation in Scratch or How to Make a Clicker Game in Scratch. Activity: ● Students will create an animation or clicker game using Scratch. ● Students must include the following kingdoms: animal, plant, fungi, protist, eubacteria, and archaea. ● Students will include the following information for each kingdom: characteristics, at least 1 "fun fact", and at least 2 examples. Wrap Up: ● Students will submit their Scratch animation or clicker game to the teacher. ● Students will share their creation with classmates to view or play. ● If time permits, students can share their creation to the class as a whole group.
Lesson links/resources:	● The 6 Kingdoms of Classification in 3 minutes ● How to Make a Clicker Game in Scratch ● How to make a presentation in Scratch **Example Projects** ● https://scratch.mit.edu/projects/220707555/ ● https://scratch.mit.edu/projects/456053811/
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions.
Time needed:	**Total time: 60 Mins** ● Warm Up: **10 Mins** ● Activity: **40 Mins** ● Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	**Teacher:** ● Computer ● Projector/smartboard with sound **Students:** ● Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.4.2** Use classification methods to explore the diversity of organisms in kingdoms (animals, plants, fungi, protists, bacteria). Support claims that organisms have shared structural and behavioral characteristics.

Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch
Scratching the Surface of Newton's Laws	
Lesson overview:  	**Purpose:** ● Students will create an animation discussing all aspects of Newton's Law. **Lesson:** Warm-Up: ● Review Newton Law's with the class as a whole group. ● Review the Scratch website and how to create an animation. Activity: ● Students will create a Scratch animation to provide information on Newton's Law

	• Each law should be defined, with an example. • Students should utilize various tools, such as Sprites, backdrops, events, sounds, etc. Wrap-Up: • Students will submit their Scratch animation or clicker game to the teacher. • Students will share their creation with classmates to view or play. • If time permits, students can share their creation to the class as a whole group.
Lesson links/resources:	• How to make a presentation in Scratch • scratch.mit.edu **Example Projects** • https://scratch.mit.edu/projects/181039882/
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**P.6.6.1** Use an engineering design process to create or improve safety devices (e.g., seat belts, car seats, helmets) by applying Newton's Laws of motion. Use an engineering design process to define the problem, design, construct, evaluate, and improve the safety device*
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

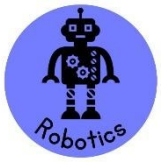
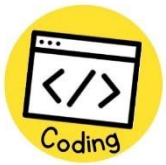
Energy and Motion	
Lesson overview:  	**Purpose:** ● Students will create a program that will provide information about potential energy, kinetic energy, and thermal energy, as well as how they are connected. **Lesson:** Warm Up: ● Review potential energy, kinetic energy, and thermal energy, as well as how they are connected. ● Watch the "Introduction to Sprite Lab" video. ● This activity can be completed individually, or in small groups. Activity: ● Students will log-on to code.org and open a new Sprite Lab that they will rename to "Energy and Motion". ● Instruct students to watch the Sprite Lab: Introducing Sprite Lab Video.

	• Students will create a program that will provide information about the potential energy, kinetic energy, and thermal energy, as well as how they are connected. • Students will utilize the world tool to set an appropriate background. (There is a larger selection of backgrounds if you click "set background to" and click the down arrow. At the bottom of the background menu students will see "more". They can click that and have access to more backgrounds. • Students will utilize the sprites tool for characters and objects. (There is a larger selection of sprites if you click "Make new..." and click the down arrow. At the bottom of the sprite menu students will see "costumes" they will click that. Then they will click "New Costume" and select something from the costume's library, draw their own, or upload and image. • Students will utilize tools such as events, behaviors, loops, variables, text, etc. to generate an informative sprite lab that explains each type of energy, as well as their relationships to one another. Wrap Up: • Students will share their presentations with the class.
Lesson links/resources:	• Sprite Lab • Introduction to Sprite Lab Tutorial
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**P.6.6.7** Determine the relationships between the concepts of potential, kinetic, and thermal energy.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

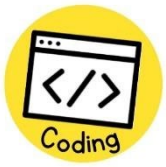
Solar System	
Lesson overview:  	**Purpose:** ● Students will create a Scratch program that will compare characteristics and movements of planets in our solar system. **Lesson:** Warm Up: ● Review the solar system and the movements of planets. Activity: ● Students will be placed into groups of 2-4 and assigned one of the following: Sun, Mercury, Venus, Mars, Earth, Jupitar, Saturn, Uranus, Neptune, or Pluto. ● Students will research information on their assigned topic. ● Each group should include the following information: the distance from Earth (if it is Earth, then distance from the sun), the diameter, how many moons it contains, period of revolution, period of rotation, and include at minimum 3 fun facts. ● Each group should incorporate the following in their scratch program: sprites, backdrop, sound, and "say" functions. ● Students should document where they found the information for their project. Wrap Up: ● Students will share their presentations with the class.
Lesson	● Scratch

links/resources:	• Introduction to an Interactive Presentation • Build Your Own Presentation **Example Projects** • The Solar System - Example Project • Solar System - Example Project
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**E.6.8.4** Obtain and evaluate information to model and compare characteristics and movements of objects in the solar system (including planets, moons, asteroids, comets, and meteors.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch and CS First - Google

Moon Phases	

Lesson overview:  	**Purpose:** ● Student will program a robot to reach correct destinations on a grid depicting/ labeling the phases of the moon. **Lesson:** Pre-Lesson Prep Work: ● The teacher will create a grid 5 x 5 grid. (This grid can be taped out on a table or the floor for a codable robot, or it can be taped out on the floor to use a student as the robot.) There should be a clearly defined start square at the top, left corner of the grid. ● The teacher will print pictures of the earth, moon phases, and sun to use in the activity. Warm Up: ● Review moon phases with the class. ● Students can complete this activity individually, or in small groups. Activity: This activity can be implemented in two ways: <u>Moon Phase Images</u> ● To begin, you can have a blank grid that only has the start square, Earth, and sun depicted. ● The teacher will show students an image of a moon phase, and the students must provide the program to place the image in the correct position. *Example: If the teacher showed an image of a full moon, then I would program my robot in the following manner:* *Forward, turn right, forward, forward.* <u>Labeling Moon Phases</u> ● The teacher will have the moon phases placed in the correct positions on the grid surrounding the earth. ● The teacher will call out a moon phase, and the students will have to program the robot to reach the correct image. *Example: If the teacher called out full moon, then I would program my robot in the following manner:* *Forward, turn right, forward, forward.* Wrap Up: ● Students will submit their grids and programming to the teacher as a formative assessment.
Lesson links/resources:	● Moon Phases Robot Grid ● Printable Arrows
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions.
Time needed:	**<u>Total time:</u> 60 Mins** ● Warm Up: **10 Mins** ● Activity: **40 Mins** ● Wrap Up: **10 Mins** (This lesson will require a full 60 minutes to complete)
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound

	● Moon Phases Robot Grid ● Printable Arrows Students: ● Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**E.6.8.6** Design models representing motions within the Sun-Earth-Moon system to explain phenomena observed from the Earth's surface (positions of celestial bodies, day and year, moon phases, solar and lunar eclipses, and tides.)
Vocabulary:	
Notes:	
Cell Explorations	

Lesson overview:  	**Purpose:** ● Students will create an animation that reviews the parts of different cells. **Lesson:** Warm Up: ● The teacher will introduce the lesson by reviewing with the students the components and their purposes of a certain type of cell (i.e. animal, plant, etc.). Activity: ● Students will start a new program in Scratch. Students will load an image of an unlabeled cell diagram as the backdrop (see links in resources for options). ● Students will program their sprite to travel to each part of the cell diagram when that part is clicked. ● When a component is clicked, the component name will appear, and the Sprite will describe its function/purpose to the user. Cell component names will remain on the screen until all components have been clicked. *(Another option is to have students program their Sprite to travel to all the cell components in a particular order and at each, the name and purpose will appear. This option removes the user input of clicking but instead will run the cell diagram the exact same way every time the green flagged is clicked.)* ● Students should provide instructions for users on how to use their program in the instructions box. Wrap Up: ● Students will test out another student's cell program to ensure it operates as intended and provide a peer review. **Extension:** ● *Have students alter their program to be interactive where the user can type in the name of a cell component and the sprite will go to that component on the screen and tell the purpose. This will require students to learn about user inputs, variables, and sensing.*
Lesson links/resources:	● Scratch ● Blank animal cell diagram: option 1 ● Blank animal cell diagram: option 2 ● Blank plant cell diagram ● Blank bacteria cell diagram
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. **AP.2.8a** Students will test programs by considering potential errors, such as what will happen if a user enters invalid input. **AP.2.10a** Students should provide documentation for end users that explains their artifacts and how they function.
Time needed:	**Total time: 60 Mins** ● Warm Up: **10 Mins** ● Activity: **40 Mins** ● Wrap Up: **10 Mins** ● (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	Teacher: ● Computer

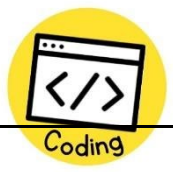
	• Projector/smartboard with sound • Blank cell template Students: • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.1.3** Develop and use models to explain how specific cellular components (cell wall, cell membrane, nucleus, chloroplast, vacuole, and mitochondria) function together to support the life of prokaryotic and eukaryotic organisms to include plants, animals, fungi, protists, and bacteria (not to include biochemical function of cells or cell part).
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

2023

	Native American Star Quilts - (Code.org) **Star Quilts Module: Lesson 1**	
	Lesson overview:  	**Purpose:** ● In this Unplugged lesson, students will be introduced to Native American Star Quilts and their significance. **Lesson:** Warm Up: ● "Star Quilt" video Activity: ● Graphic Organizer and coloring utensils ● Students will make a 6-pointed star shape on paper. One of the main goals is to discuss the patterns within the shape. These ideas will be key for the online coding portions of the activity later. Wrap-up: ● Debrief
	Lesson links/resources:	● Code.org Native American Star Quilts Lesson 1
	CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. For example, when programming an interactive story, students could use a compound conditional within a loop to unlock a door only if a character has a key AND is touching the door. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. For example, animations can be decomposed into multiple scenes, which can be developed independently.
	Time needed:	**Total time: 45 Mins** ● Warm-Up: 10 minutes ● Main Activity: 25 minutes ● Wrap-Up: 10 minutes
	Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound ● Star Quilt Article - Resource ● Star Quilts - Slides (Download) ● Star Quilts - Video Students: ● Star Quilts Worksheet - Handout ● Virtual Pattern Blocks (Optional) - Resource
	Subject integrated:	Social Studies, Math
	Other standards addressed:	**H.6.1** Explain the characteristics and development of culture. **H.6.1.1** Describe the major aspects of culture (religion/belief systems, language, ethnicity, institutions, technology, art, architecture, dress, foods, traditions, etc.). **H.6.1.2** Explain how culture changes as it is passed from one generation to the next.

		H.6.1.3 Identify major culture regions of the world and explain how the characteristics of each set it apart from the others. **6.G.1** Find the area of right triangles, other triangles, special quadrilaterals, and polygons by composing into rectangles or decomposing into triangles and other shapes; apply these techniques in the context of solving real-world and mathematical problems.
	Vocabulary:	**Pattern** - Something that happens or appears in a regular and repeated way
	Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

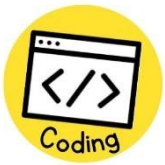
Getting Started with The Artist - Code.org
Star Quilts Module: Lesson 2

Lesson overview: 	**Purpose:** ● This skill-building lesson provides an opportunity for students to become familiar with the Artist Tool and to build important coding skills. Students learn how the stamping tool works, how to turn x degrees in an angle, and how to use loops.

	• Students will create a simple star shape. By the end of the levels, students will manipulate angle measurements to see the effect on the number of points on the star. Computers allow for accuracy and precision in studying shapes and patterns because images are generated in a matter of seconds instead of relying on hand-drawn shapes. **Lesson:** Warm Up: • What patterns do you see, what patterns can you think exist in the world around us and discuss vocabulary. Activity: • Students go through skill building lessons to create patterns. Wrap Up: • Debrief
Lesson links/resources:	• Code.org Native American Star Quilts Lesson 2
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. For example, when programming an interactive story, students could use a compound conditional within a loop to unlock a door only if a character has a key AND is touching the door. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. For example, animations can be decomposed into multiple scenes, which can be developed independently.
Time needed:	**Total time: 50 Mins** • Warm Up: 10 minutes • Main Activity: 30 minutes • Wrap Up: 10 minutes
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Star Quilts - Slides (Download) Students: • Computer/tablet with internet access • Virtual Pattern Blocks (Optional) - Resource
Subject integrated:	Social Studies, Math
Other standards addressed:	**H.6.1** Explain the characteristics and development of culture. **H.6.1.1** Describe the major aspects of culture (religion/belief systems, language, ethnicity, institutions, technology, art, architecture, dress, foods, traditions, etc.). **H.6.1.2** Explain how culture changes as it is passed from one generation to the next. **H.6.1.3** Identify major culture regions of the world and explain how the characteristics of each set it apart from the others. **6.G.1** Find the area of right triangles, other triangles, special quadrilaterals, and polygons by composing into rectangles or decomposing into triangles and other shapes; apply these techniques in the context of solving real-world and mathematical problems.

Vocabulary:	**Artist** - You can write code to make him draw almost anything. **Loop** - A sequence of code that is repeated. **Point** - One part of the star shape
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Code Your Star Quilt
Star Quilts Module: Lesson 3

Lesson overview:  	**Purpose:** ● In the final lesson for this activity, students are ready to build their own Star Quilt. ● Students will build one point of the star and then loop their code to end up with an 8 point traditional Star Quilt. Star Quilts are meant to be gifted so students can share their project with someone special. **Lesson:** Warm Up: ● Follow the prompts on the Code.org Native American Star Quilts Lesson 3 Lesson Plan ● What is the difference between a 6-pointed star shape and an 8-pointed Star Quilt? Activity:

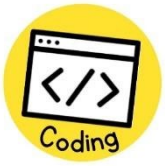
	• Star Quilts – Skill Building Wrap Up: • Reflection and Sharing
Lesson links/resources:	• Code.org Native American Star Quilts Lesson 3 Lesson Plan
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. For example, when programming an interactive story, students could use a compound conditional within a loop to unlock a door only if a character has a key AND is touching the door. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. For example, animations can be decomposed into multiple scenes, which can be developed independently.
Time needed:	**Total time: 50 Mins** • Warm-Up: 10 minutes • Main Activity: 30 minutes • Wrap-Up: 10 minutes
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Star Quilts - Slides (Download) • Virtual Pattern Blocks - Resource Students: • Computer/tablet with internet access
Subject integrated:	Social Studies, Math
Other standards addressed:	**H.6.1** Explain the characteristics and development of culture. **H.6.1.1** Describe the major aspects of culture (religion/belief systems, language, ethnicity, institutions, technology, art, architecture, dress, foods, traditions, etc.). **H.6.1.2** Explain how culture changes as it is passed from one generation to the next. **H.6.1.3** Identify major culture regions of the world and explain how the characteristics of each set it apart from the others. **6.G.1** Find the area of right triangles, other triangles, special quadrilaterals, and polygons by composing into rectangles or decomposing into triangles and other shapes; apply these techniques in the context of solving real-world and mathematical problems.
Vocabulary:	**Comments** - Notes to yourself that explain a section of code. They do not alter the program but help keep your code in order.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Designing for Accessibility	
Lesson overview: 	**Purpose:** ● Through learning about accessibility, students recognize the impacts of computing beyond their own lives. Accessibility might not seem like a relevant CS topic but creating technology that is accessible for underserved users helps make tech better for everyone else as well. **Lesson:** Warm Up: ● Follow the prompts on the Code.org Designing for Accessibility Lesson Plan ● What is an app? Why have apps become so popular? Activity: ● Designing for Accessibility Wrap Up: ● Reflection and sharing
Lesson links/resources:	● Code.org Designing for Accessibility Lesson Plan
CS standards addressed:	**CS 2.1** Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices. **CS2.1.a** Make recommendations for existing devices (e.g., a laptop, phone, or tablet) or design their own components or interface (e.g., create their own controllers). T

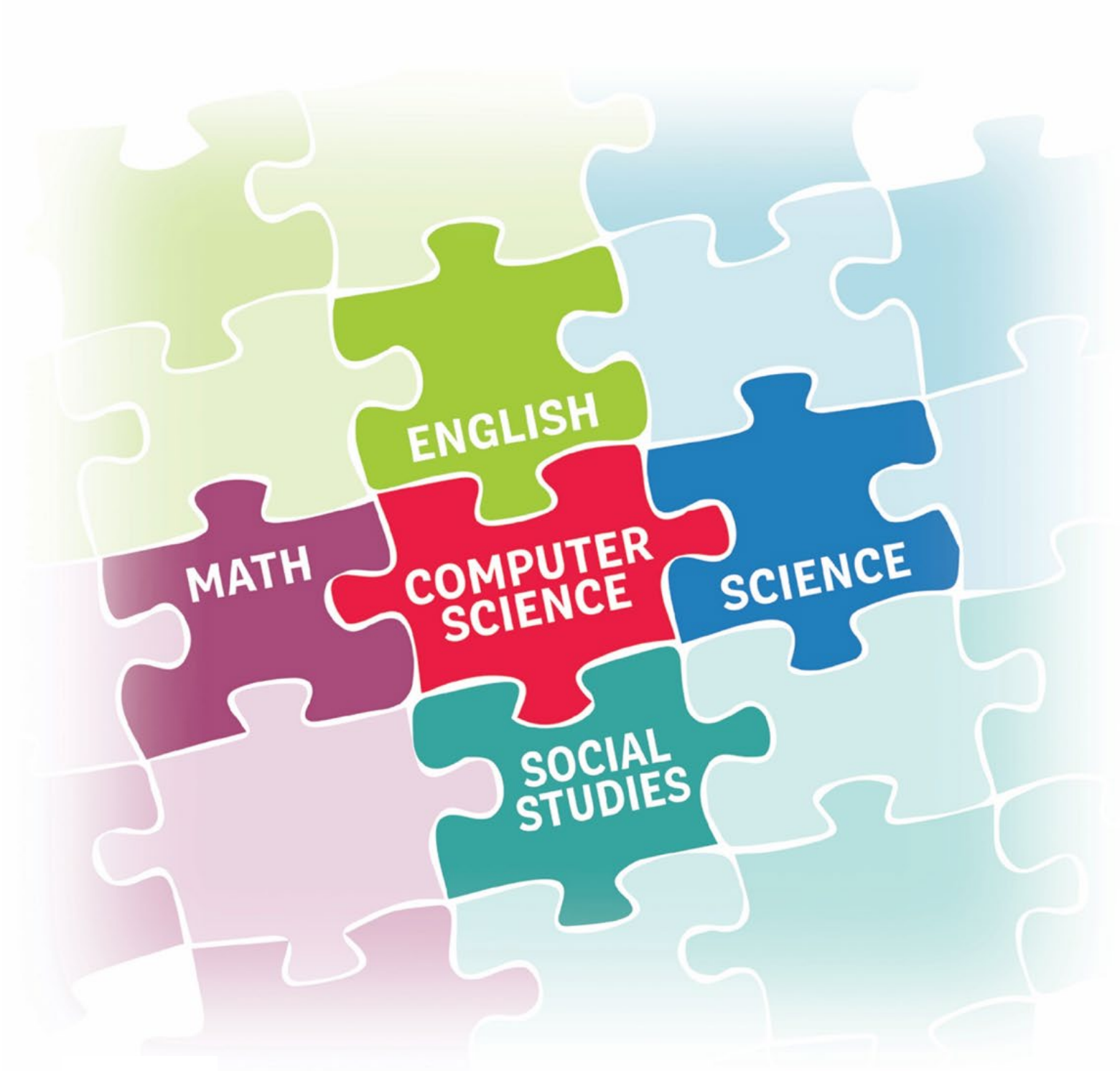
Time needed:	**Total time: 50 Mins** • Warm Up (10 min) • Activity (35 min) • Wrap Up (5 min)
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Types of Disabilities - Resource Students: • Designing for Accessibility - Slides • crayons/pencils • Journals
Subject integrated:	Social Studies
Other standards addressed:	**G.6.7** Compare and contrast ways that humans and the physical environment are impacted by the extraction of resources. **G.6.7.3** Describe examples of how the physical environment provides opportunities and constraints for human activities.
Vocabulary:	**Accessibility** – the extent to which a service, device, or product is usable by as many individuals as possible, including people who have disabilities. **Empathy** – being able to know how someone is feeling, even when you aren't in the same situation.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Finding Credible News: How do we find credible information on the internet?	
Lesson overview: 	**Purpose:** ● Help students dig into why and how false information ends up online in the first place, and then practice evaluating the credibility of what they're finding online. **Lesson:** This activity can be completed individually, or in small groups. Warm Up: ● Follow the prompts on the Common Sense Education: Finding Credible News Lesson Plan ● "Tricky Wiki" Analyze: ● "News or Fake News?" Wrap Up: ● "Fighting Fake"
Lesson links/resources:	● Common Sense Education: Finding Credible News Lesson Plan
CS standards addressed:	**IC2.4** Describe tradeoffs between allowing information to be public and keeping information private and secure. **IC.2.1a** Students should consider current events related to broad ideas, including privacy, communication, and automation.
Time needed:	**Total time: 45 minutes** ● Warm Up: 10 mins ● Analyze: 25 mins ● Wrap Up: 10 mins

Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Lesson Slides • News or Fake News? • Example #3 Article Handout (Student Version) • Lesson Quiz Students: • Computer/tablet with internet access Take Home Resources: • Family Activity • Family Tips • Family Engagement Resources
Subject integrated:	Social Studies
Other standards addressed:	**CI.6.2** Examine the challenges of civic engagement in the contemporary world. **CI.6.2.1** Compare the positive and negative impacts of changing technologies on expanding the role of citizens throughout the world and the challenges posed by new media sources to obtaining reliable information upon which to make decisions.
Vocabulary:	**Credible** – able to be believed. **Corroboration** – an additional source that confirms or supports a news story, article, or piece of information. **Bias** – showing a strong opinion or preference for or against something or someone. **Evaluate** – to carefully examine something to figure out its value.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at https://www.commonsense.org/education

Cybersecurity - Simple Encryption	
Lesson overview:  	**Purpose:** ● "Encryption" is a process for transforming a message so that the original is "hidden" from anyone who is not the intended recipient. Encryption is not just for the military and spies anymore. We use encryption every day on the Internet, primarily to conduct commercial transactions, and without it our economy might grind to a halt. ● This lesson gives students a first taste of the kind of thinking that goes into encrypting messages in the face of computational tools. Computational tools dramatically increase the strength and complexity of the algorithms we use to encrypt information, but these same tools also increase our ability to crack an encryption. Developing strong encryption relies on knowledge of problems that are "hard" for computers to solve and using that knowledge to encrypt messages. As a resource, you may wish to read all of Chapter 5 of Blown to Bits. It provides social context which you may want to bring to your classroom. **Lesson:** Warm Up: ● Follow the prompts on the Code.org Cybersecurity - Simple Encryption Lesson Plan ● Classic Encryption – Caesar Cipher Activity: ● Crack a Caesar Cipher ● Crack a Random Substitution Cipher Wrap Up: ● "Encryption and Public Keys" video ● Discussion ● Career discussion
Lesson links/resources:	● Code.org Cybersecurity - Simple Encryption Lesson Plan

CS standards addressed:	**NI.2.3a** Students should encode and decode messages using a variety of encryption methods, and they should understand the different levels of complexity used to hide or secure information.
Time needed:	**Total time:** ● Warm Up: 10 mins ● Main Activity: 35 mins ● Wrap Up: 15 mins
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound Students: ● Computer/tablet with internet access
Subject integrated:	Social Studies
Other standards addressed:	**H.6.1.1** Describe the major aspects of culture (religion/belief systems, language, ethnicity, institutions, technology, art, architecture, dress, foods, traditions, etc.).
Vocabulary:	**Caesar Cipher** - a technique for encryption that shifts the alphabet by some number of characters. **Cipher** - the generic term for a technique (or algorithm) that performs encryption. **Cracking encryption** - When you attempt to decode a secret message without knowing all the specifics of the cipher, you are trying to "crack" the encryption. **Decryption** - a process that reverses encryption, taking a secret message and reproducing the original plain text. **Encryption** - a process of encoding messages to keep them secret, so only "authorized" parties can read it. **Random Substitution Cipher** - an encryption technique that maps each letter of the alphabet to a randomly chosen other letters of the alphabet.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

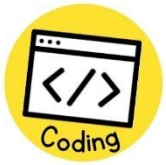


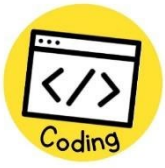
2023

Elementary Integration Guide

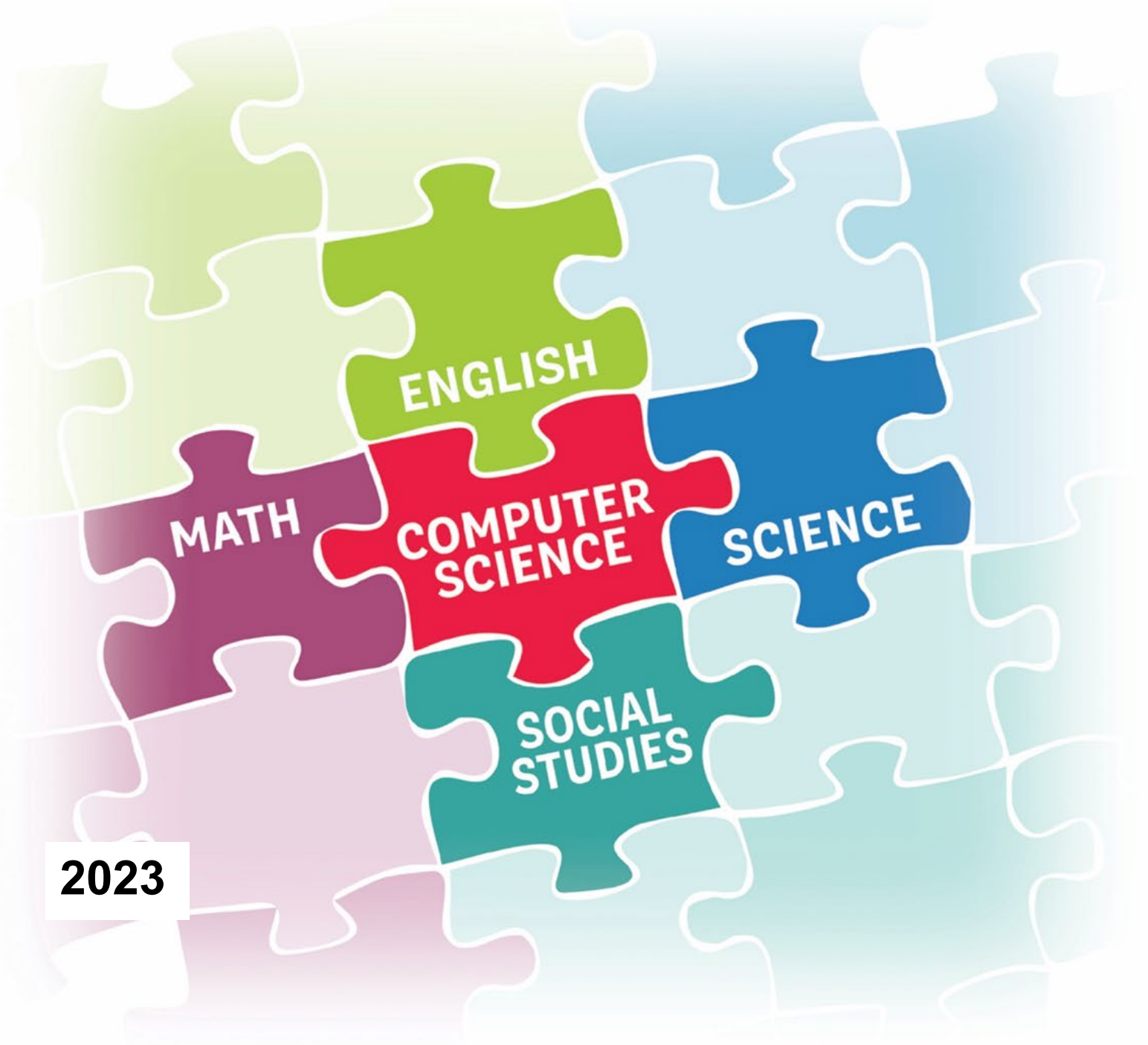
SOCIAL EMOTIONAL

“All About Me” Animation

Lesson overview:  	**Purpose:** ● Students will create an animation on Scratch describing their talents, skills, and other positive attributes. **Lesson:** Warm Up: ● What are talents? What are skills? What are positive attributes? Activity: ● Students will make a list, on paper, of their own talents, skills, and other positive attributes. ● Students will use Scratch to insert a Sprite and animate a list or story of their talents, skills, and other positive attributes. The students can build their animation to best represent them (i.e., Sprite choice, background, etc.). Wrap Up: ● Reflection and sharing with classmates.
Lesson links/resources:	● https://scratch.mit.edu/
CS standards addressed:	**AP.2.7a** Students should use portions of code, algorithms, and/or digital media in their own programs and websites
Time needed:	**Total time: 60 Mins** ● Warm Up: 10 Mins ● Activity: 40 Mins ● Wrap up: 10 Mins
Materials needed:	Teacher: ● Teacher/Student Scratch accounts (optional) Students: ● Paper and pencil ● Internet connected device
Subject integrated:	SEL
Other standards addressed:	**1B.** Develop an accurate perception of oneself (i.e., beliefs, values, skills, talents, and interests) **1B.8** Identify positive attributes and qualities about oneself including talents, interests, physical characteristics, etc. **1B.9** Describe characteristics that are important to oneself (i.e., loyalty, honesty, etc.) **1B.10** Describe how one’s personal qualities, interest, beliefs and academic/career goals impact decision making.
Vocabulary:	**Talents** – The natural ability to do something better than most people. **Skills** – ability that comes from training or practice. **Positive Attributes** – qualities, character traits, and strengths that are considered good or help us in some way.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

Conflict Resolution	
Lesson overview:  	**Purpose:** ● Students will learn about conflict and conflict resolution. Students will create a conflict and proper resolution in a Scratch animation. **Lesson:** Warm up: ● Discuss the meaning of conflict and conflict resolution. Provide scenarios for the students to understand. ● This activity can be completed individually, or in pairs. Activity: ● Student(s) to work to create a Scratch animation depicting a conflict (provided by the teacher) and the appropriate resolution. Wrap Up:

	• Have students share their stories with classmates.
Lesson links/resources:	• https://scratch.mit.edu/
CS standards addressed:	**AP.2.7a** Students should use portions of code, algorithms, and/or digital media in their own programs and websites
Time needed:	**Total time: 60 Mins** • Warm Up: 10 mins • Activity: 40 mins • Reflection and sharing: 10 mins
Materials needed:	Teacher: • Teacher/Student Scratch accounts (optional) Students: • Paper and pencil • Internet connected device
Subject integrated:	SEL
Other standards addressed:	**4C.** Demonstrate the ability to successfully manage and resolve conflict in relation. **4C.11.** Identify behaviors that create conflict (e.g., spreading rumors, inappropriate posts or texts on social media, wrongful accusations, and insult or put downs). **4C.12.** Apply conflict resolution skills in order to de-escalate, defuse and resolve a conflict. 13. Determine strategies for avoiding or resolving conflicts related to destructive peer pressure.
Vocabulary:	**Conflict** – a challenge to the way a person thinks or behaves. **Conflict Resolution** – the process that two or more people use to agree on a solution to a problem. **De-escalate** – to lessen they intensity or anger in a conflict.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch



2023

Finding Balance in a Digital World

Lesson overview:  	**Purpose:** ● Reflect on their common online and offline activities. ● Identify ways to "unplug" to maintain balance between online and offline activities. ● Use the Digital Habits Checkup routine to create a personal challenge to achieve more media balance. **Lesson:** Warm Up: ● Follow the prompts on the Common Sense Education: Finding Balance in a Digital World Lesson Plan ● "It's a Digital World!" Reflect: ● "My Online and Offline Life" Apply: ● "Balance It Out" Wrap Up: ● "Finding Media Balance"
Lesson links/resources:	● Common Sense Education: Finding Balance in a Digital World Lesson Plan
CS standards addressed:	**IC.2.4** Describe tradeoffs between allowing information to be public and keeping information private and secure.
Time needed:	**Total time: 60 Mins** ● Warm Up: 10 mins ● Reflect: 10 mins ● Apply: 15 mins ● Wrap Up: 15 mins ● Quiz: 10 mins
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound Students: ● Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**W.6.4** Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. **W.6.8** Gather relevant information from multiple print and digital sources; assess the credibility of each source; and quote or paraphrase the data and conclusions of others while avoiding plagiarism and providing basic bibliographic information for sources.
Vocabulary:	**digital media** – Information that comes to us through the internet, often through a table, smartphone, or laptop. **media balance** – using media in a way that feels healthy and in balance with other life activities (family, friends, school, hobbies, etc.)

	red flag feeling - when something happens on digital media that makes you feel uncomfortable, worried, sad, or anxious. **unplug** - to engage in activities that don't involve devices, apps, or the internet. digital habits - behaviors we do often or regularly with digital media and devices
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Common Sense Education
Don't Feed the Phish	
Lesson overview: 	**Purpose:** ● Compare and contrast identity theft with other kinds of theft. ● Describe different ways that identity theft can occur online. ● Use message clues to identify examples of phishing. **Lesson:** Warm Up:

	• Follow the prompts on the Common Sense Education: Don't Feed the Phish Lesson Plan • "Safe or Unsafe?" Explore: • "How Identity Theft Happens" Analyze: • "How to Catch a Phish" Wrap Up: • "Stay Safe from Scams"
Lesson links/resources:	• Common Sense Education: Don't Feed the Phish Lesson Plan
CS standards addressed:	**IC.2.4** Describe tradeoffs between allowing information to be public and keeping information private and secure.
Time needed:	Total time: 60 Mins • Warm UP: 10 mins • Explore: 15 mins • Analyze: 15 mins • Wrap Up: 5 mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**W.6.4** Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. (Grade-specific expectations for writing types are defined in standards 1–3 above.) **W.6.9** Draw evidence from literary or informational texts to support analysis, reflection, and research.
Vocabulary:	**Private information** – information about you that can be used to identify you because it is unique to you (e.g., your full name or your address). **Identify theft** – a type of crime in which your private information is stolen and used for criminal activity. **Phishing** – when someone poses as an institution, like a bank or school, and sends you a personalized message asking you to provide private information. **Internet scam** – an attempt to trick someone, usually with the intention of stealing money or private information. **Shortened URL** – a web address that has been condensed and which could mislead a user into going into a risky website.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Common Sense Education

“Who Are You Online?”	
Lesson overview:   	**Purpose:** ● What does it mean to "be yourself" or to "be "real"? Those are deep thoughts for any middle schooler. For kids today, these questions matter online, too. Help your students explore why some people create different or alternate personas for themselves online and on social media. ● Reflect on reasons why people might create fake social media accounts. ● Identify the possible results of posting from a fake social media account. ● Debate the benefits and drawbacks of posting from multiple accounts. **Lesson:** Warm Up: ● Follow the prompts on the Common Sense Education: "Who Are You Online?" Lesson Plan

	• Why "Finstas"? Explore: • "Which Me Should I Be?" Debate: • "The Finsta Debate"
Lesson links/resources:	• Common Sense Education: "Who Are You Online?" Lesson Plan
CS standards addressed:	**NI.2.2a** Students will explain how physical and digital security measures protect electronic information. **IC.2.4a** Students should discuss and describe the benefits and dangers of allowing information to be public or kept private and secure.
Time needed:	**Total time: 45 Mins** • Warm Up: 10 Mins • Which Me Should I Be: 15 Mins • The Finsta Debate: 20 Mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**W.6.1** Write arguments to support claims with clear reasons and relevant evidence. **W.6.1a** Introduce claim(s) and organize the reasons and evidence clearly. **W.6.1b** Support claim(s) with clear reasons and relevant evidence, using credible sources and demonstrating an understanding of the topic or text. **W.6.1c** Use words, phrases, and clauses to clarify the relationships among claim(s) and reasons.
Vocabulary:	**Anonymous** – without a name or other information that identifies who you are. **Affinity Group** – a group of people linked by a common interest or purpose. **Curate** – to select, organize and look after a collection (e.g., content posted to a social media profile) **Finsta** – a fake Instagram account used for posting to a specific group of people or to post anonymously
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Common Sense Education

Chatting Safely Online	
Lesson overview: •	**Purpose:** ● Games, social media, and other online spaces give kids opportunities to meet and chat with others outside the confines of their real-life communities. But how well do kids know the people they're meeting and interacting with? Help students consider whom they're talking to and the types of information they're sharing online. ● Analyze how well they know the people they interact with online. ● Reflect on what information is safe to share with different types of online friends. ● Learn to recognize red flag feelings and use the Feelings & Options thinking routine to respond to them. **Lesson:** Warm Up: ● Follow the prompts on the Common Sense Education: Chatting Safely Online Lesson Plan ● Who You're Talking to Online Evaluate: ● Two Online Chats

	Analyze: ● Red Flag Feeling Wrap Up: ● Exit Ticket
Lesson links/resources:	● Common Sense Education: Chatting Safely Online Lesson Plan
CS standards addressed:	**IC.2.4a** Students should discuss and describe the benefits and dangers of allowing information to be public or kept private and secure.
Time needed:	**Total time: 60 Mins** ● Warm Up: 10 mins ● Evaluate: 15 mins ● Analyze: 15 mins ● Wrap Up: 10 mins ● Discuss "Take-Home" Resources: 10 mins
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound Students: ● Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**W.6.4** Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. (Grade-specific expectations for writing types are defined in standards 1–3 above.) **W.6.10** Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of discipline-specific tasks, purposes, and audiences.
Vocabulary:	**Private Information** – information about you that can be used to identify you because it is unique to you (e.g., your full name or your address). **Red Flag Feeling** – when something happens on digital media that makes you feel uncomfortable, worried, say, or anxious. **Inappropriate** – not acceptable in the situation; not okay **Risky** – potentially harmful to one's well-being
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Common Sense Education

Digital Drama Unplugged	
Lesson overview:   	**Purpose:** ● Miscommunication is a common occurrence online and on social media. Plus, being behind a screen makes it easier to say things they wouldn't say in person. So how do we help students avoid the pitfalls of digital drama? Help them learn tips on avoiding online drama in the first place and de-escalating drama when it happens. ● Reflect on how easily drama can escalate online. ● Identify de-escalation strategies when dealing with digital drama. ● Reflect on how digital drama can affect not only oneself but also those around us. **Lesson:** Warm Up: ● Follow the prompts on the Common Sense Education: Digital Drama Unplugged Lesson Plan ● "What is Digital Drama?" Explore: ● "Where do People Stand?" Wrap Up: ● "Act It Out"
Lesson links/resources:	Common Sense Education: Digital Drama Unplugged Lesson Plan

CS standards addressed:	**IC.2.1a** Students should consider current events related to broad ideas, including privacy, communication, and automation.
Time needed:	**Total time: 60 Mins** ● Warm Up: 5 mins ● Explore: 25 mins ● Wrap Up: 15 mins ● Discuss "Take-Home" Resources: 15 mins
Materials needed:	Teacher: ● Computer ● Projector/smartboard with sound Students: ● Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**W.6.8** Gather relevant information from multiple print and digital sources; assess the credibility of each source; and quote or paraphrase the data and conclusions of others while avoiding plagiarism and providing basic bibliographic information for sources. **W.6.9** Draw evidence from literary or informational texts to support analysis, reflection, and research.
Vocabulary:	**De-escalate** – to lessen they intensity or anger in a conflict. **Digital Drama** – when people use devices, apps, or websites to start or further a conflict between people.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Common Sense Education

Appendix A: Code.org

I'd like to start using Code.org in my classroom. How should I start?

https://support.code.org/hc/en-us/articles/228116468-I-d-like-to-start-using-Code-org-in-my-classroom-How-should-I-start-

How to create a teacher account:

https://support.code.org/hc/en-us/articles/228116468-I-d-like-to-start-using-Code-org-in-my-classroom-How-should-I-start-

How to create a classroom section:

https://support.code.org/hc/en-us/articles/115000488132-Creating-a-classroom-section

Finding curriculum and lesson plans:

https://support.code.org/hc/en-us/articles/115001595051-Finding-curriculum-and-lesson-plans

Code.org Support

https://support.code.org/hc/en-us

Appendix B: Scratch

SCRATCH

Educator's Guide

- Teacher Accounts
- Beginner's Guide
- Lesson Guides