Coding Whisperers

Educator Guide

Math and Coding

3-5

Welcome to CreositySpace

We're glad you're here.

We know our curriculum might look a little different from what teachers are used to, but it was developed with the philosophy that students learn better when what they are learning is put into a context that is relevant to them.

With a focus on student-led, inquiry-based discovery, each unit features a variety of real-life STEM entrepreneurs, their personal stories, and details about the businesses they are building and technologies they are developing. With that as our anchor, CreositySpace connects the applications back to the relevant elementary-level science topics and creates lessons that can be delivered in science, ELA, math, social studies, and art classes.

How do we do this?

CONNECT students and their interests, the entrepreneurs and their technologies, and the relevant standards through an overarching phenomenon.

ENGAGE students through a variety of introductory activities designed to spark student interest and excitement about the unit and give teachers a chance to assess where students are with their thinking, interests, and misconceptions.

TRANSFORM students' self-confidence and proficiency with science as they build foundational knowledge through the early lessons and investigations and then apply and demonstrate their deeper understanding of key concepts through design projects, models, and summative challenges.

Since one can never predict the exact path a student's mind will explore, the CreositySpace team is always available to provide additional support and content should your students' questions take you down an unfamiliar road.

Integrating STEM and CreositySpace into Your Classroom

The CreositySpace *Educator Guide* is your resource to engage your classroom in student-directed inquiry-based discovery. You decide what sequencing works best for your classroom. We provide the tools and content designed to leverage your students' creativity and curiosity, with the extra Background Information and Additional Resources so that you are supported wherever their inquiries take you.

To help with lesson planning options, the Lesson Planning Tools and Pacing Guide Resources sections are there to help you determine preparation activities, lesson flow, and assessments. The design projects and summative challenges provide an opportunity for your students to demonstrate their knowledge in a variety of ways.

Informational Icons

<table>
<tr>
<td>

Student Questions

The inquisitive monkey represents the portion of the lesson where students should be given the opportunity to voice or revisit their questions.

</td>
<td>

Student Sense Making

The jubilant tiger represents the portion of the lesson where students should take time to reflect upon and make sense of what they have observed and learned.

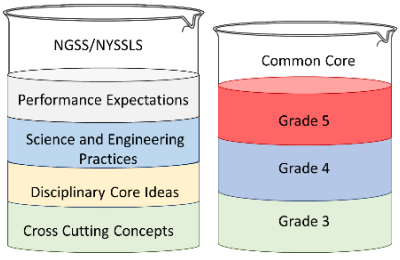
</td>
</tr>
<tr>
<td>

Time Management

To support lesson planning, stopwatch icons indicate how much time an activity or investigation is expected to take.

</td>
<td>

Standards



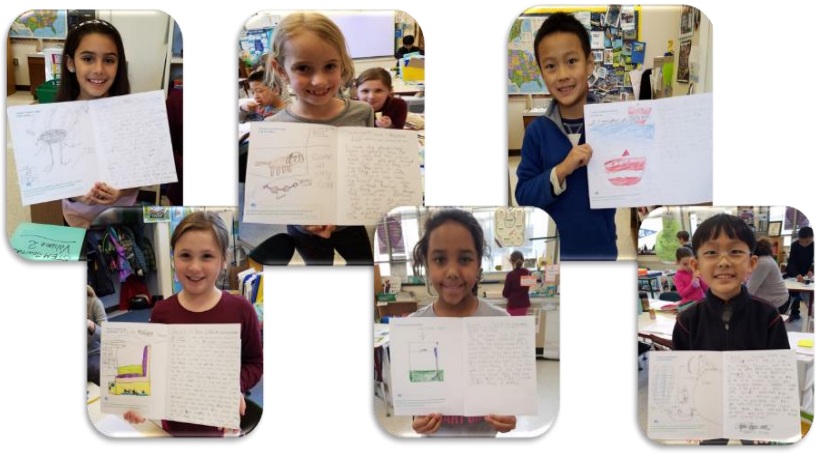
Multicolored beakers highlight the Common Core (CC) and NGSS/NYSSLS standards supported by a given plug-in, investigation, or challenge.

</td>
</tr>
</table>

Using the *Book of Ideas* to Create an Environment of Creativity, Communication, and Collaboration in Your Classroom

Getting kids to connect to and see value in their ideas is a key ingredient in teaching STEM and to fostering student confidence. This is why CreositySpace created the *Book of Ideas*.

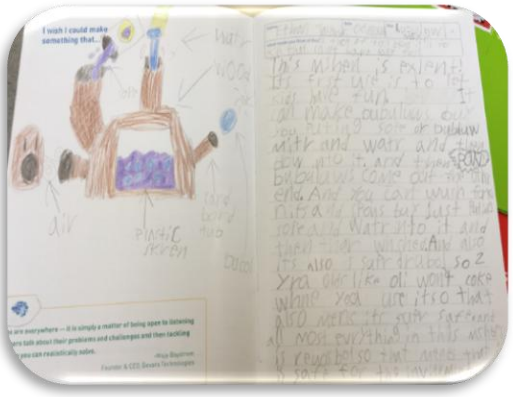
Similar to a real inventor's notebook, the *Book of Ideas* was designed to encourage kids to write down or draw, explore, and discuss their own ideas and inventions. Pages include a title block, quotes from entrepreneurs and innovators, short design prompts and a list of inventions and businesses started by kids.

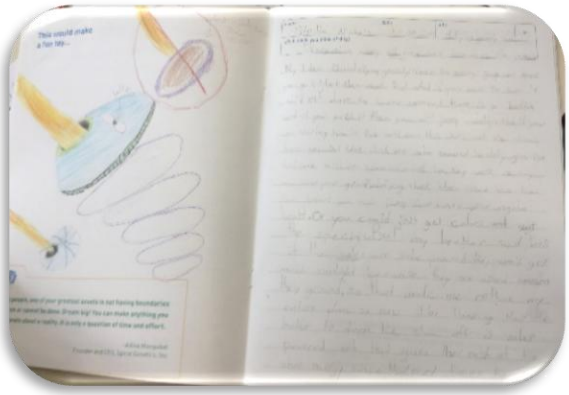


The *Book of Ideas* is NOT about the right answer, perfect spelling, or fitting into a pre-determined box.

The *Book of Ideas* IS about

- Giving students the opportunity to explore and express their ideas in a way that works for them,
- Providing a forum in which students may work independently, as well as collaboratively, and practice giving and receiving respectful feedback, and
- Offering an exciting way for students to show you the ideas, big wonderings, and solutions on their minds and giving you additional insight on how to engage every learner.





There is no wrong way to use the *Book of Ideas* in your classroom. Above all, it should be considered a conversation-starter and a great way to encourage creativity, communication, collaboration, and curiosity in your classroom.

Assessment and Differentiation

Assessments

This Educator Guide provides a number of pre, formative, and summative assessments as well as ongoing opportunities for student self and peer assessment. The Appendix also contains an outline for the range of acceptable work for many of the formative assessments (e.g., exit tickets, short research activities, etc.) and expected prior knowledge for the specific investigations (see the *My STEM Stories*™ and *My STEM Explorer Notes*™ sub-sections). A checklist or detailed rubric are provided for the models and summative assessments respectively.

It is not required that students complete ALL provided assessments, as there is redundancy built in to give you the flexibility to match assessments to your students' interests and learning styles. However, if your students are struggling with a particular learning objective or concept, unused assessments provide an opportunity for additional practice or alternative perspectives.

Differentiation

Group work and peer discussion is encouraged as a strategy to support students with language and learning challenges. However, all students are expected to maintain their own records in their *My STEM Stories*™ and *My STEM Explorer Notes*™ notebooks so that educators can assess the learning of individual students. Drawing and the use of one's native language are acceptable methods to maintain records and demonstrate scientific knowledge. The use of technology to assist challenged learners (e.g., video recording, voice-to-text, Google Translate, etc.) is encouraged if available.

The Detailed Description portion in the Main Investigation section describes extensions for more advanced students. These extensions are in bold blue text.

Ongoing Support

Successful implementation of each CreositySpace unit is important to us, and to that end, our company is committed to providing ongoing support to you—from brainstorming ideas and helping with an activity to answering questions around implementation. Don't hesitate to reach out to us via email or phone.

We hope *Coding Whisperers* inspires and energizes your classroom to explore the intersection of science with the world around you, and we welcome your feedback on what you like, would like to see, or even change. Feel free to reach out to us at Peg@CreositySpace.com or Kath@CreositySpace.com.

Table of Contents

**Next Generation Science Standards (NGSS) is a registered trademark of WestEd. Neither WestEd nor the lead states and partners that developed the standards were involved in the production of, and does not endorse, this product.*

Introduction

Coding Whisperers: Content Connections

Outlined below is the progression of learning objectives for the *Coding Whisperers* unit. The theme of the coding and physical computing invites you and your students on an introductory exploration of the strengths, and limitations, of computer programming.

Please make sure to go through the Before You Begin section under Main Investigations to ensure the hardware and software components are properly setup.

Learning Progression

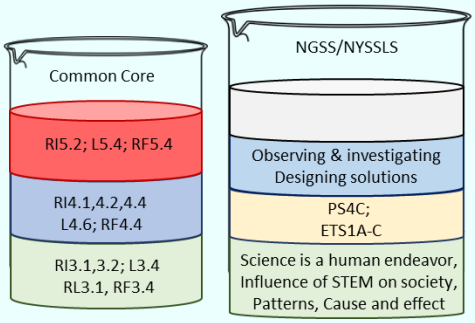
CONNECT students and their interests, the entrepreneurs and their technologies, and the relevant standards through the overarching phenomenon: *How do you create a video game?*

ENGAGE students with the **Robot Chef** introductory challenge, the **Introduction to Scratch** coding activity, and discussions about some of the innovative ways entrepreneurs are using computer programming to help communities around the world.

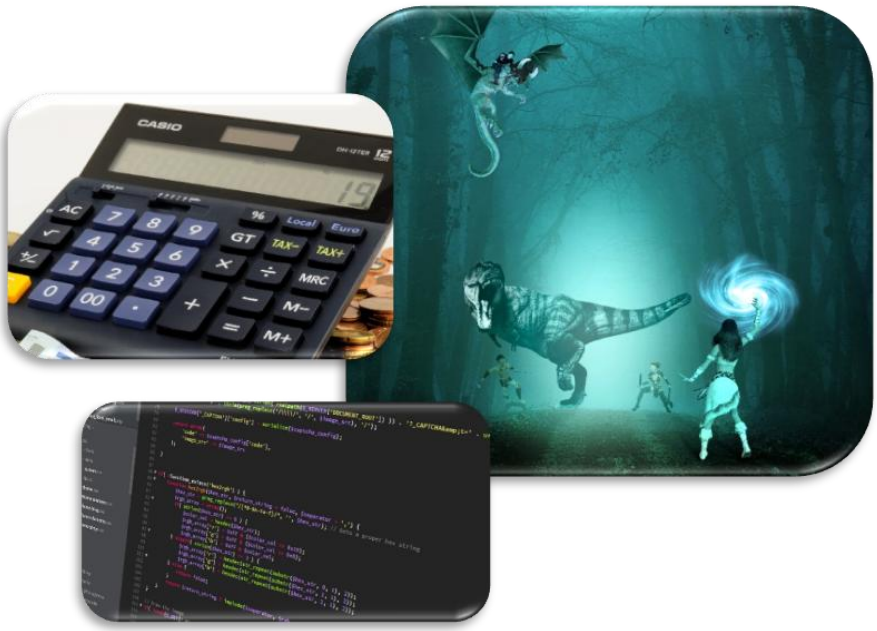
TRANSFORM students' self-confidence and proficiency as they build foundational knowledge about coding through the **Animation, Introduction to the BBC micro:bit,** and **Can You Hack Flappy Bird?** investigations.

Students apply and demonstrate their deeper understanding of computer programming through their **Design Your Own Video Game** design challenge and **Computing Solutions** summative challenge.

Technology Description: Coding

Time	*Standards*	*Description*
		*The introduction section was initially developed to give you, the teacher, some additional context on the scientific field and focus surrounding the highlighted technologies in this unit. However, we realized that this was also good background and informational text reading for the students. This text, along with a few reading comprehension questions, is included in the* My STEM Stories™ *notebooks.*

Coding, also known as computer programming, is the language of electronics. It can be as straightforward as asking a device to calculate two plus two or as complicated as creating a video game world where the number of outcomes between two players is almost impossible to quantify. Coding is such a big part of our everyday life that soon babies might learn a computer language at the same time they are learning how to speak!



There are many examples of coding all around us. One example is physical computing, which is when computers and electronics interact with the outside world through a series of rules or preprogrammed commands. One example you might be familiar with is a video game controller. With physical computing, the electronics respond to specific instructions that are written in the code. These instructions may come from the words in the code itself or how the code tells the electronics to respond to an external action (like moving a joystick or removing a light source).

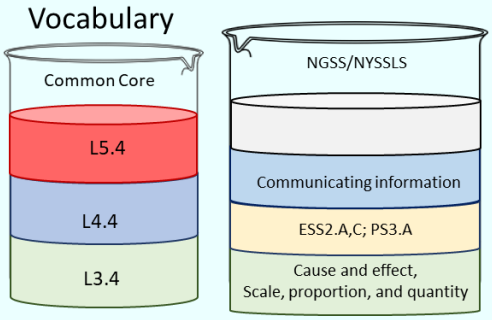
Another field of coding is that of data analysis—which is paving the way for the development of new technologies such as artificial intelligence. In these applications computers analyze a variety of inputs and then make their own decisions. These decisions can be a simple as determining the pattern in a series of numbers or as complicated as a self-driving car or a chess-playing robot.

Every day entrepreneurs, engineers, and regular people (like you and me) come up with new ways to use coding to solve problems in their lives. Is there a challenge in your life that you think coding could help you solve?

Vocabulary: Computer Science

Time

Standards



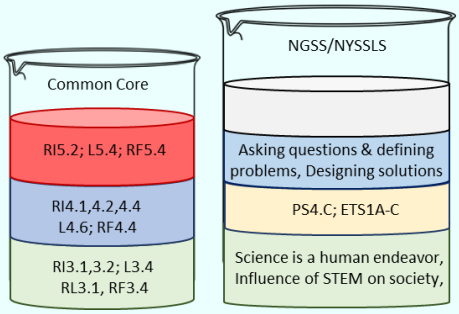
Description

The table below contains key vocabulary words for this unit specifically related to the entrepreneur and application. The My STEM Stories™ *notebook contains the vocabulary table with the Term and Definition columns completed and a blank third column that encourages students to determine a picture or simplified definition. Columns 3 and 4 in the table below are intended to provide you with some examples of simplified definitions or appropriate pictures. (Note: Drawing diagrams is a skill needed in higher level sciences.) Additional unit vocabulary is provided in the Background Information section.*

Term	Definition	Simpler Definition	A Picture or Simplified Definition
Computer Programming (coding)	Computer programming is a way for a person to translate their logic into a computer language.	Giving directions to a computer in a special language	Telling the computer what to do
Computer Hardware	Computer hardware are the parts of a computer you can touch—such as the screen, computer chips, and keyboard.	Parts of a computer you can touch	
Computer Software	Computer software, or computer code, is the set of instructions that tell the computer what to do.	The instructions that tell the computer what to do	

Physical Computing	Physical computing is the ability to control computer hardware and physical devices (motors, sensors, lights) with computer software.	Giving instructions to computer hardware	
Algorithm	An algorithm is a guide, or an outline, that is used to lay out all the steps needed for a computer program or code.	General outline for a piece of software	
Input/Output (I/O)	Inputs and Outputs allow for the user to control software. Touchscreens, keyboards, mice, and joysticks are all examples of I/O devices.	Parts of a computer that send your messages into the brain of the computer	

Your Technology: Physical Computing

Time	*Standards*	*Description*
		*This section introduces the entrepreneurs, technologies, and businesses that form the anchor applications and phenomena for this unit. These minibiographies of real people developing real technologies make the elementary concepts covered in this unit current, real, and relevant for the students and answer the "Why are we learning this?" question before it is even asked. Text and additional reading comprehension questions are also provided in the* My STEM Stories™ *notebooks.*

Enduring Understandings

Students will learn the basics of computer programming, physical computing, and how they can be used in a variety of applications.

Emma Yang and Timeless (http://www.timeless.care/)

Emma Yang started coding when she was only six and began developing the Timeless app when she was only 8-years old.

At first, she focused her coding efforts on making simple video games but one day she noticed things didn't seem quite right with her grandmother, who lived in Hong Kong. It started when her grandmother forgot her birthday. Then she started calling the house over and over again not realizing they had spoken on the phone a few minutes earlier. It was around this time that Emma's family discovered her grandmother had Alzheimer's disease—a disease that effects your brain and makes it hard to remember names, faces, recent activities, or upcoming plans. Emma wanted to help her grandmother, so she started to do some research.

She figured out what the hardest tasks were for people with Alzheimer's and learned what they needed most. In addition to help with remembering things, one of the most important needs for Alzheimer's patients is that they stay connected with their family and community. In 2019, she

launched the Timeless app on the App Store. Timeless focused on keeping people connected while addressing some of the most common challenges. Some other features of the app included daily reminders, photo grouping, and recognition (using artificial intelligence), as well as alerts for things that might be forgotten such as repeated phone calls.

Recently, as Emma started college, she realized she needed to step back from working on and promoting Timeless to focus on school. As a result, there was no one to continue updating the Timeless code and she decided to remove it from the App Store. Despite this transition for her business, Emma remains passionate about using technology to support our aging community and the Timeless website remains an active record of her contribution to this field. Listen to Emma talk about why she developed the Timeless app.

- https://www.youtube.com/watch?time_continue=91&v=hR8I66p66mU A 3:45 min video about the app.
- https://www.youtube.com/watch?v=olZTPRbXGw4 A 4:10 min video about Emma and what motivates her.
- https://www.youtube.com/watch?v=wOKJ0-jOSyY A 30 min+ interview/talk by Emma at Google.

Alexander Deans and the iAid Device

Alexander Deans, a 16-year-old from Ontario, Canada, developed the iAid device to help people who are blind or have trouble seeing get around more easily. He designed a joystick device that can be hooked up to your phone and helps the user move around without hitting things. Listen to Alex describe his innovation in this video www.youtube.com/watch?v=EGPo7gnvlhE&t=1s.

This is a wonderful TEDx talk by Alex. It is a bit long (9 min) but very rich in content to build the confidence of young learners that it is worth the time.
https://www.ted.com/talks/alex_deans_life_lessons_of_a_teen_inventor?language=en

Unearth Technologies

Meet Nate, Amy, and Brian. They started a company called Unearth Technologies. They each had different skills and interests. Brian liked business and management experience. Amy had software and product development experience. Nate was focused on visual design and understanding customers.

Computers, cell phones, satellites and drones can collect a lot of information—so much information that it can be hard to find what you need (like finding matching socks in a huge pile of laundry!). That's where Unearth Technologies made a difference. They were experts in both COLLECTING and ORGANIZING information. Unearth Technologies designed special software to help the construction industry to help them keep track how far along projects were and where everyone and everything was.

In 2023 Unearth Technologies was acquired by Procore and Nate, Amy and Brian moved on to their next adventure. Congratulations to all of them!

Roy Allela and Sign-IO

Roy Allela is from Kenya. He studied engineering and computer science in college. His 6-year-old niece is deaf, and he wanted to help her communicate. She knows sign language, but many people cannot understand sign language.

Roy developed a glove that his niece could wear that had sensors in each finger. These sensors can tell how much the finger was bent and how it was moving. When the gloves are paired with a cell phone or tablet,

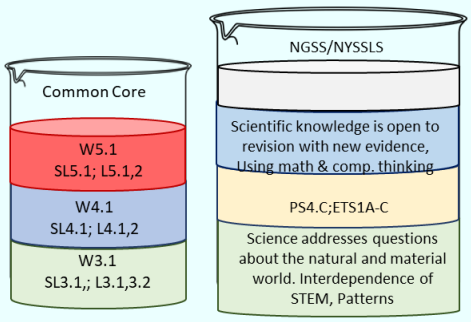
much the same way we pair wireless headphones, and the phone or tablet turns the motion into words. Then, the phone or table can “speak” using a text-to-speech program.

Although Sign-IO gloves are not perfect, they are a great tool for people, especially younger people, who need help with everyday communication.

Technology Historical Timeline: Numbers and Computing

Time

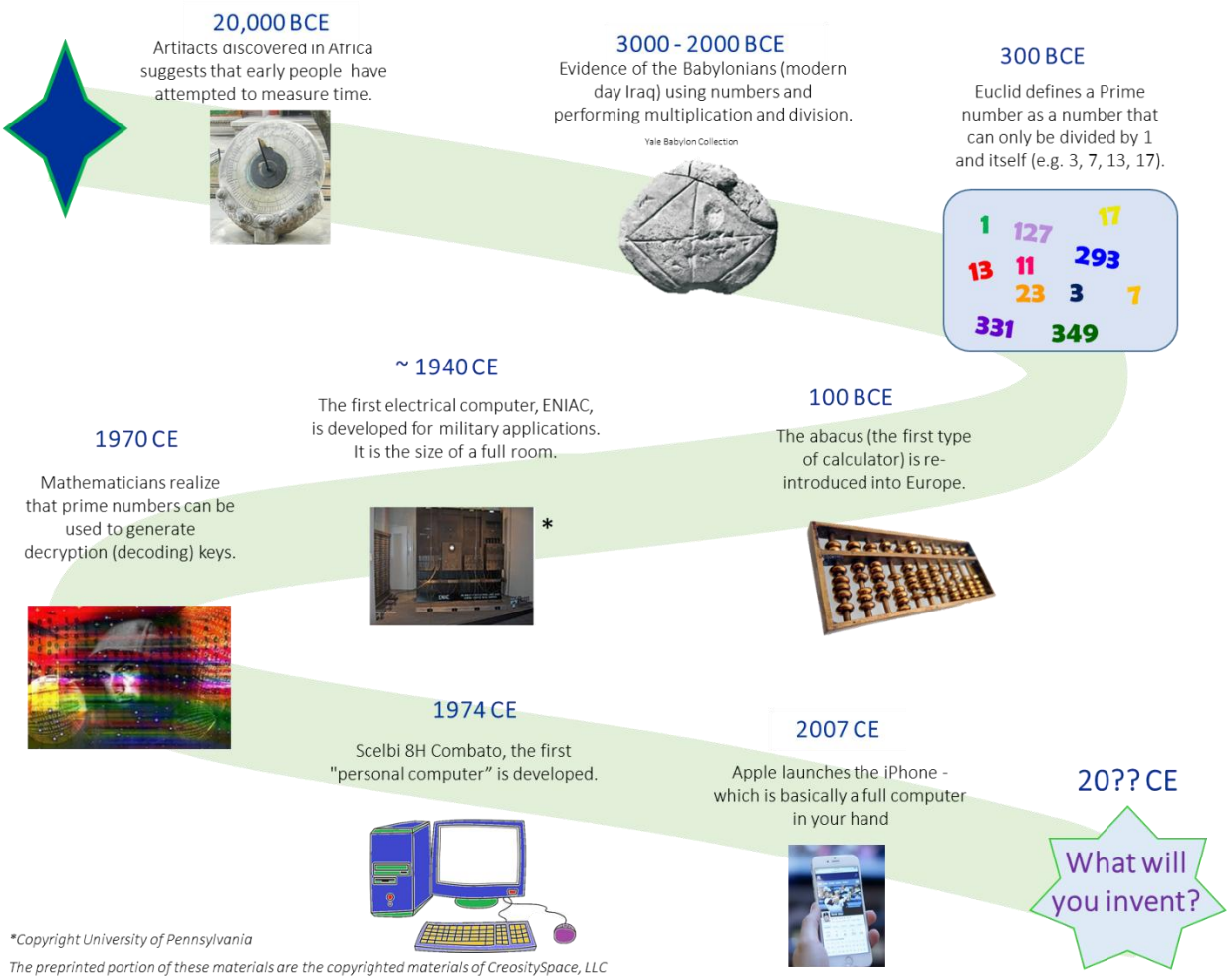
Standards



Description

The Technology Historical Timeline is a great tool to use to give your students historical context for what they are learning and to show them how any single scientific discovery or understanding is built from all the discoveries and understandings that came before. In many cases, scientific discoveries only thrived if there was a community need they helped to solve.

The Appendix contains a variety of timelines and suggested activities you can use with your class. Many of these activities are a good opportunity for ***peer-to-peer*** *and* ***teacher-student feedback*** *cycles.*



Background Information

This section contains additional information intended for teachers on the topics of:

- The engineering design process
- Computers, microprocessors, and coding
- Coding languages
- Kit components and system set-up

Links to additional videos, articles, webpages, etc., are provided in the Additional Resources section where they are organized by topic area. Extra copies of the printed materials can be found in the Appendix.

The Engineering Design Process

Despite what the name might suggest, the **engineering design process** is really no different than any creative or iterative process. You would follow the same basic steps if you were writing a story or painting a picture.

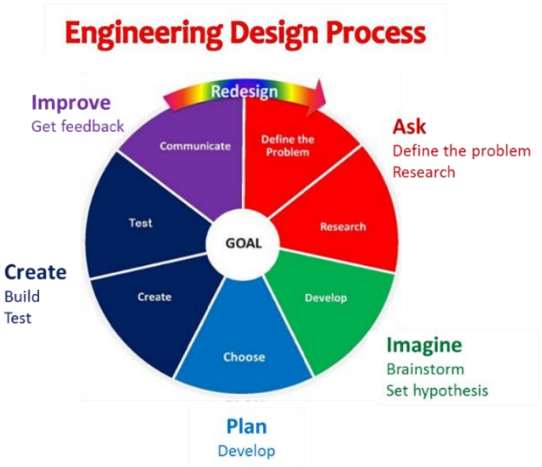
Step 1: Start with a question, problem, or goal.

Step 2: Think about all the possibilities.

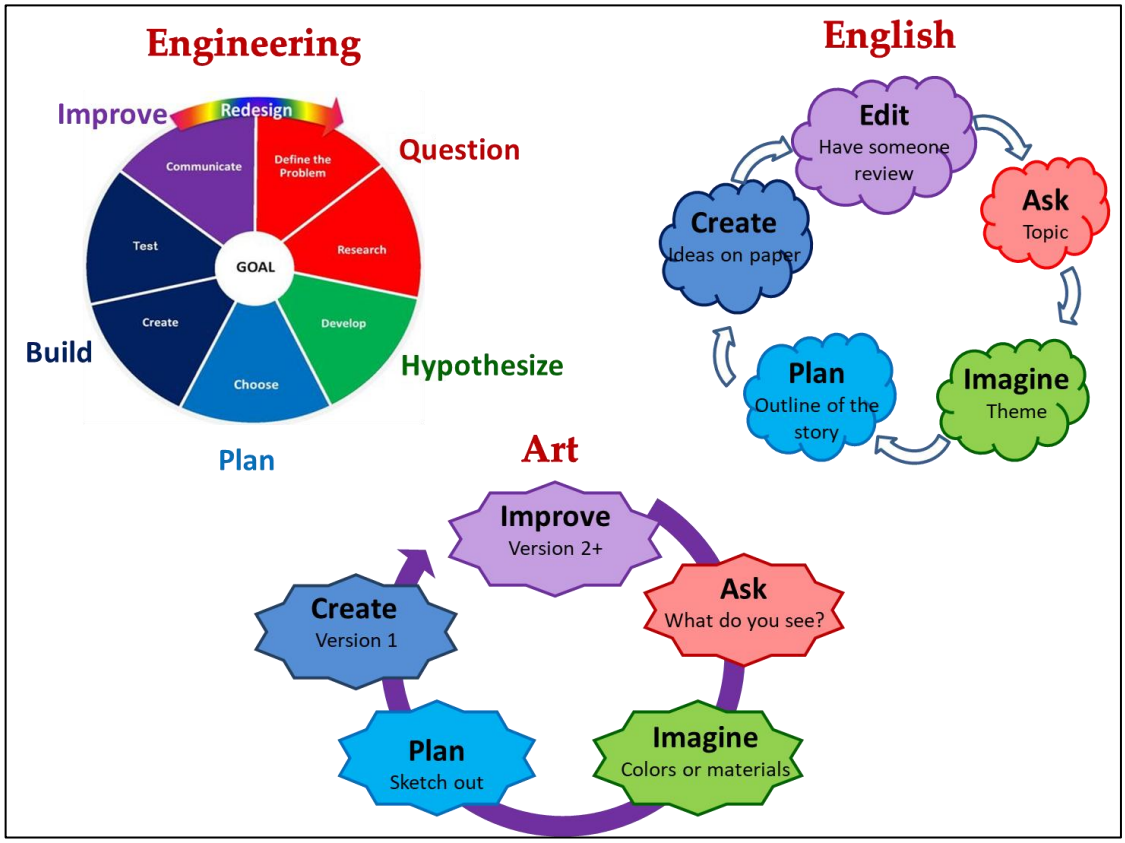
Step 3: Decide which ideas from step 2 you want to use.

Step 4: Create your first draft/prototype/version.

Step 5: Get feedback and improve your design.



Under some circumstances the words used to describe a step might be different, but the general goals of each step are the same.



Additional Vocabulary

Brainstorming	Brainstorming is when a group of people gets together and shares ideas to solve a problem or challenge (coming up with a lot of ideas).
Prototype	A prototype is an early version of an invention. It can't necessarily do everything the invention is supposed to do, but it allows the inventor to see what things will look like.

Computers, Microprocessors, and Coding

So much of what we do every day involves some sort of computer or programmed device. Even though almost all of us use them, many of us rarely have a chance to stop and think about what is going on inside.

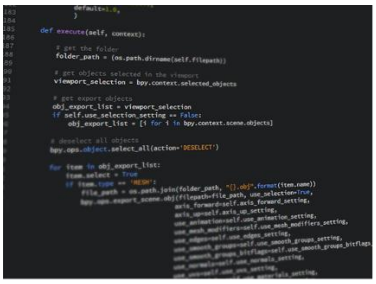
<table>
<tr><td colspan="3">Computers and Programmable Devices
 
This is the broad category of all things programmable.
  </td></tr>
<tr><td colspan="2">Hardware
Hardware is the part of the programmable device that you can touch (although that doesn’t mean you should).</td><td>Software
Software are the rules or instructions given to the programmable device. You cannot touch software.</td></tr>
<tr><td>External Hardware
The external hardware are the portions of the computer/programmable device you are expected to touch (controllers, keyboards, cases, etc.)
  </td><td>Internal Hardware
The internal hardware is the portion of the computer/programmable device that you shouldn’t touch (microprocessors (a.k.a. computer chips), contacts, resistors, etc.)
  </td><td>Just like spoken languages, software/coding languages have both general and specific rules. The general rules are mostly around overall structure and are consistent across different coding languages. The specific rules—called syntax—are incredibly detailed rules associated with the words and symbols that should be used in the code.
  </td></tr>
</table>

Note: Common student misconception

When asked, most students will say that computers are smart. However, most computers—and other programmable devices—are not that smart (for the moment we will delay considering artificial intelligence). Most computers and programmable devices can only do what is exactly specified in the code—they can't make decisions nor can they make interpretations. The introductory activity, Robot Chef, is an engaging way to show kids this concept in action and an effective tool for correcting this misconception.

Coding Languages

As was mentioned, there are a number of different coding languages. Some of the more common examples include Python, Java, Javascript (no relation to Java), HTML, C, SQL, and PHP. Different languages have different pros and cons. Some are easy to use and can run almost anywhere (e.g., Python, Java, C) while other are best suited for a specific and involved task. For example, SQL (Structured Query Language) is used in large database systems and PHP (Hypertext Preprocessor) is often used for dynamic data heavy websites and apps.

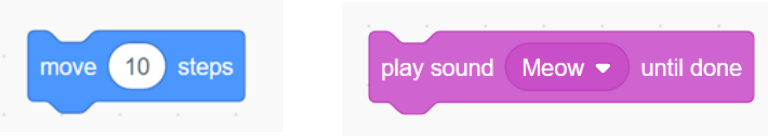


Python

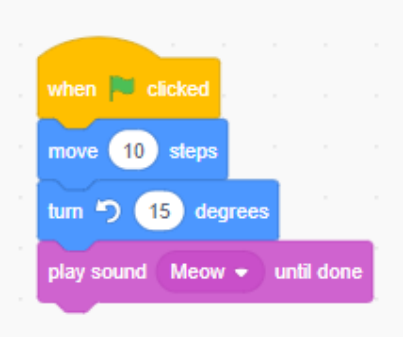
PHP

HTML

As coding has become more popular—especially with the emphasis on introducing it to younger learners—several more visual coding languages have emerged. These languages take bite-sized pieces of code and package them into an easy to use blocks—sort of like a Lego® block.



Students can drag-and-drop these blocks of code into a single workspace and "snap" them together to write a program.



In these instances, students must still consider and work through the logic of writing a computer program but don't have to worry about getting the syntax 100% correct.

Drag-and-Drop Coding: SCRATCH vs. MakeCode

Two of the more popular drag-and-drop coding languages are SCRATCH—developed by MIT—and MakeCode—developed by Microsoft. Both are good introductory coding languages for elementary students and both work well with the BBC micro:bit.

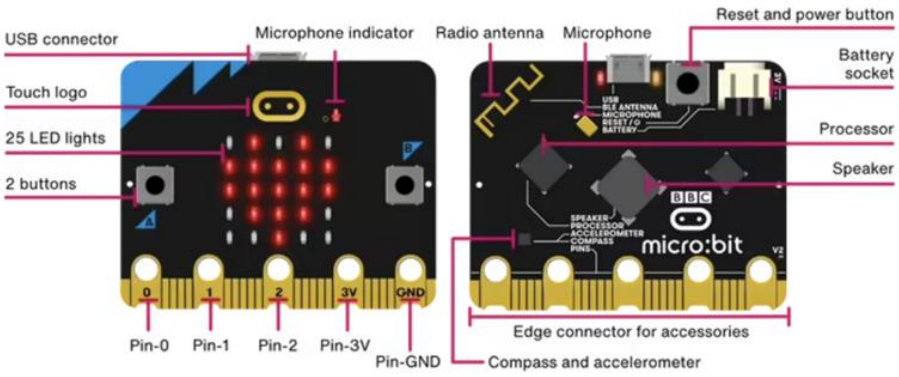
The *Coding Whisperers* unit is designed to work with SCRATCH for two key reasons:

1. The setup of the SCRATCH platform **minimizes the likelihood of confusing dead ends**. While there is no guarantee that any piece of code will work as intended, SCRATCH is generally thought to be easier to troubleshoot.
2. SCRATCH has few multifunctional coding blocks. This forces students to take fewer coding shortcuts and enables them to develop the logic and sequencing skills that will transfer more directly to more complicated coding languages.

The one downside of SCRATCH is that, given the current library of BBC micro:bit blocks, one cannot access all of the BBC micro:bit functionality, such as the light sensor or the compass. Both MakeCode and Python enable access to this additional functionality, so as your class gains experience, they may want to experiment with those programs. The BBC micro:bit website contains a number of curated tutorials for MakeCode and Python coding projects.

Kit Components and Setup

Each coding kit contains a BBC micro:bit, a micro USB connector, and four alligator clip wires.



Micro USB cable

Alligator clip wires

There are four main setup tasks that are described in detail in the Before You Begin sub-section of the Main Investigations section. We suggest you do the following things before you begin this unit. If you run into any problems please contact us at Kath@Creosity Space.com ASAP so that we can work with you to find a solution.

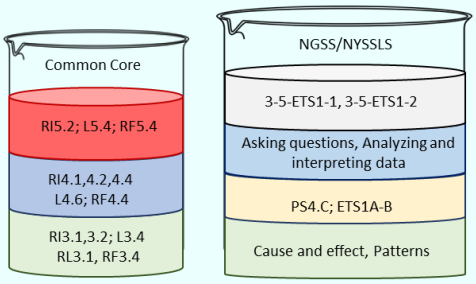
1. Ensure SCRATCH 3.0 isn't blocked by your school's firewall.
2. Download and install the BBC micro:bit SCRATCH link/app.
3. Download and install the BBC micro:bit hex code.
4. Download and save the Flappy Bird scratch project file.

Lesson Planning Tools

Topic Introduction Tools

Time

Standards



Description

The following few pages outline some topic introduction tools you can use with your students to get them excited to be thinking about coding and the community.

Pick the tools that work for you!

You do not need to use all of these introductory tools, but instead pick the ones that enable you to get a feel for what your students know and what they are interested in. These introductory tools and activities are also a good way to check in with your students throughout the unit to see how they are doing, what concepts they may be struggling with, and/or how their interests are developing.

Phenomenon

Can you create your own video game?

Essential Questions

Topical essential questions are used to help provide the "why" around each concept or standard students are learning. Also included are *big wonderings* to inspire deeper reflection and discussion.

Topical Essential Questions

- Are computers smart? What does it take to talk to a robot?
- How do I write instructions for a computer?
- What are the different pieces needed to design a video game?
- How do I design a video game?

Big Wonderings

- How can coding be used to increase help us accomplish more in a shorter amount of time as well as to give us more ways to have fun?
- How can I use coding to solve a problem in my life or community?
- Given Earth's limited resources, how can we use technology to accomplish more?

Introductory Videos

Here are a series of introductory videos on coding developed by DK books especially for younger learners. The links below take you to the YouTube videos. To avoid ads and auto play, there is a non-YouTube version of each video on the *Coding Whisperers* digital forum.

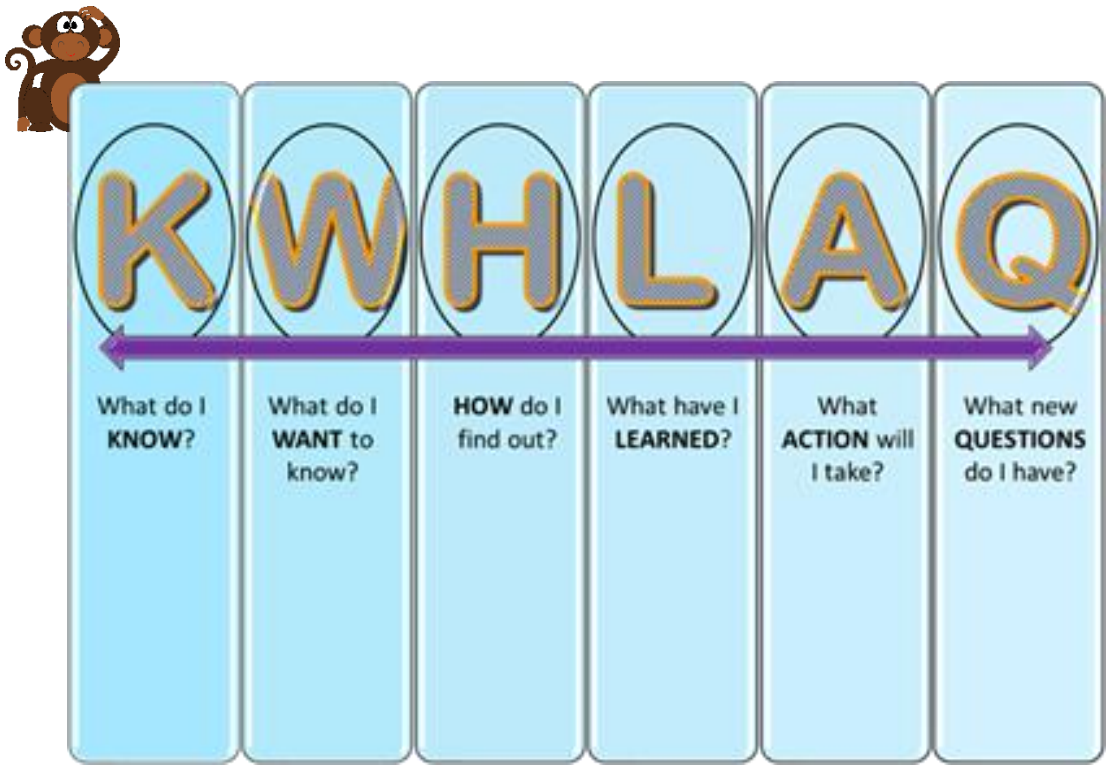
1. https://www.youtube.com/watch?v=THOEQ5soVpY 1:55 min
2. https://www.youtube.com/watch?v=Nc31NAujTkA 2:32 min
3. https://www.youtube.com/watch?v=xngWoocXYCo 2:48 min

KWHLAQ Charts

KWHLAQ charts (know, want, how, learned, action, questions) are a great way to get students thinking about a topic area *and give you a chance to collect questions that are interesting to them, see where they are with current understanding, and get a look for possible misconceptions*. Depending on your students you may choose to complete one or more of these activities with you leading the discussion or have the students work together in small groups.

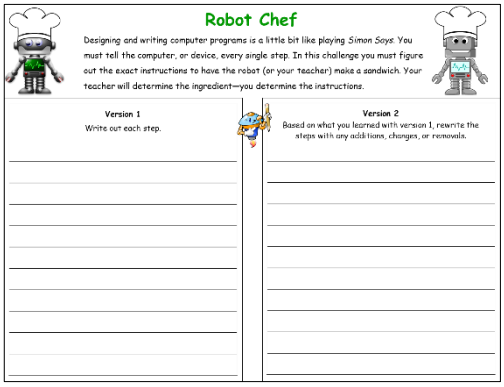
A few possible starting prompts for KWHLAQ charts are:

- What do you know about computer programming or coding?
- How often do you interact with computer programming each day?
- Are computers smart?



Introductory Investigation

The introductory investigations are designed to be activities the students can work on with very little guidance or introduction from you, the educator. The goal is that the students have a chance to think about ideas or concepts independently and in peer groups, without reliance on the adults in the room. **Your role as the teacher is to help the students stay motivated and on task, without providing them with the answer.** Often encouragement to explore their line of thinking is all that they need.



The **Robot Chef** introductory challenge gets students thinking about how specific they need to be when writing a computer program. In this challenge they must program a robot (you) to make a sandwich. If they are able to work in groups, it also enables them to work on their discussion and cooperation skills.

The sandwich can be comprised of ingredients that are appropriate for your class. Robert and Brian demonstrate this challenge about 2 minutes into the *Coding Whisperers* lesson video. **We suggest you watch that part of the video before doing this challenge with your students**. The key thing is that you must do exactly what they instruct—nothing more and nothing less.

This challenge will lay the groundwork for many of this unit's investigations and makes sure all students have some foundational experience thinking through the detailed steps of a task. Since the introductory challenge is intended to be used before the unit begins, the data recording sheet is provided separately from the *My STEM Explorer Notes*™ notebook. A full-sized copy can be found in the Appendix and on the unit website. Printed loose-leaf copies are provided in the unit kit and printed curriculum packs.

Main Investigations

The following pages describe the main investigations in this unit.

Safety

Warning: This kit contains materials that may be harmful if used incorrectly. Please read all instructions before beginning. Failure to follow these instructions and warnings could result in serious consequences.

- Kit components are scientific and engineering equipment—not toys. While we have worked to supply the most robust kit components possible, aggressive use can cause damage.
- Remind students to use materials only as instructed by the teacher.

Students must be careful not to connect the ground to the 3V pin on the BBC micro:bit

A student safety contract is provided in the Appendix in the Teacher Support Documents section.

Before You Begin
Setting Up Hardware and Software

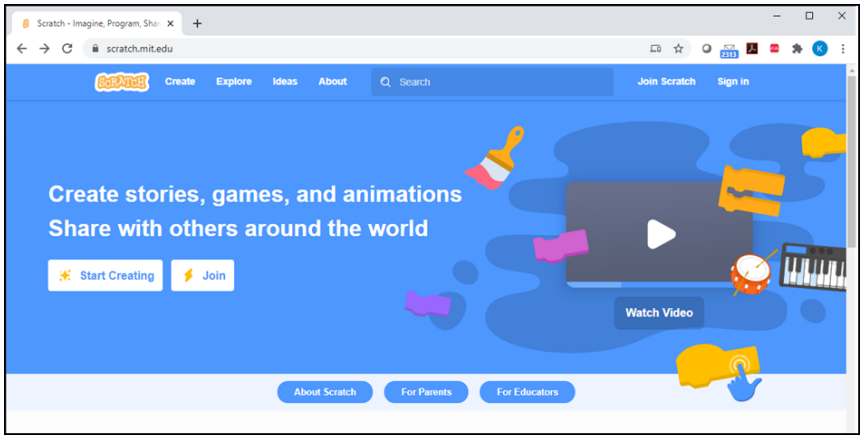
We suggest you do the following things before you begin this unit. If you run into any problems please contact us at Kath@Creosity Space.com ASAP so that we can work with you to find a solution.

1. Ensure SCRATCH 3.0 isn't blocked by your school's firewall.
2. Download and install the BBC micro:bit SCRATCH link/app.
3. Download and install the BBC micro:bit hex code.
4. Download and save the Flappy Bird scratch project file.

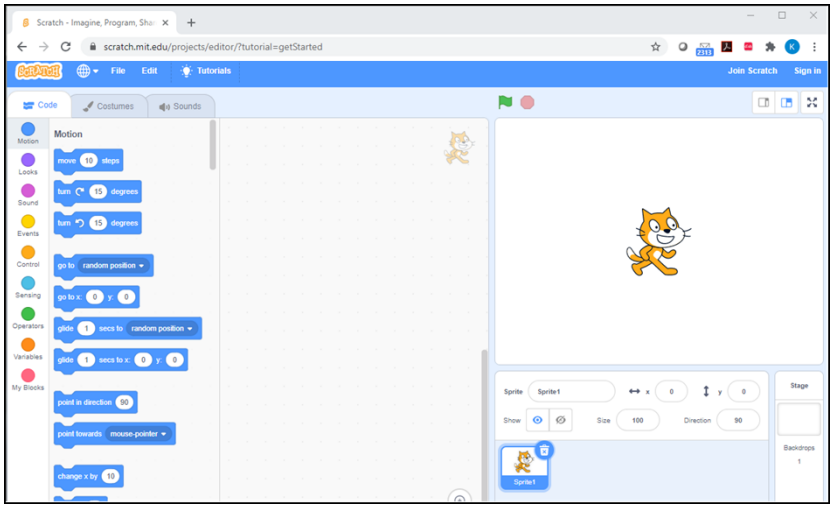
Note: Detailed instructions and links are provided in the following pages. Instructions, links, and files are also provided in the CreositySpace *Coding Whisperers* unit digital forum.

Ensure SCRATCH 3.0 isn't blocked by your school's firewall.

1. Go to https://scratch.mit.edu/



2. Select Create



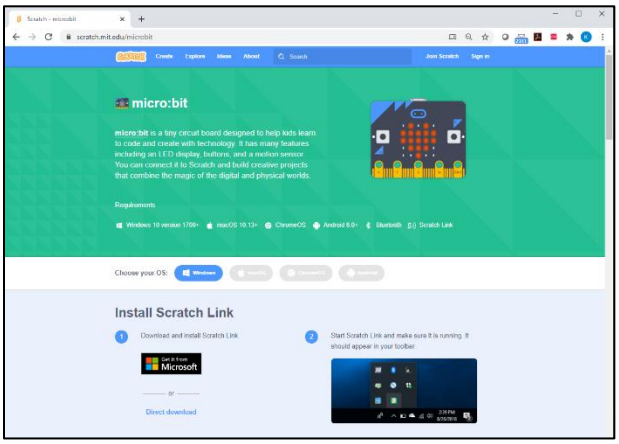
3. Begin coding

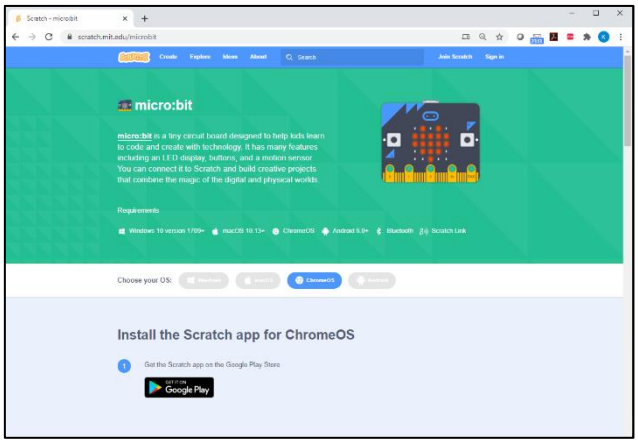
Students do not need to create an account nor download any software to begin coding with SCRATCH. They can also save their projects (code) directly to the computer (File/Save to your computer) and then reload any saved files (File/Load from your computer) without an account. If they would like to share or save their projects to the web, they will need to create an account.

Download and install the BBC micro:bit SCRATCH link/App.

To enable scratch to work with the BBC micro:bit you must install the SCRATCH link/app.

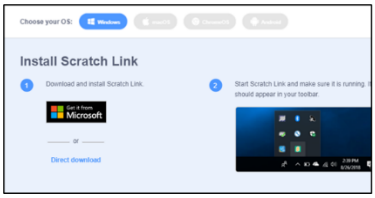
1. Go to the BBC micro:bit section of Scratch https://scratch.mit.edu/microbit and select your operating system (Windows, macOS, ChromeOS, Android). For Windows and macOS there is a link you must download. For ChromeOS and Android there is and app to download.





2. Get and install the SCRATCH link or app.

   i. For Windows and iOS operating systems you will continue to run SCRATCH through your web browser. You will need to make sure the SCRATCH Link is running.



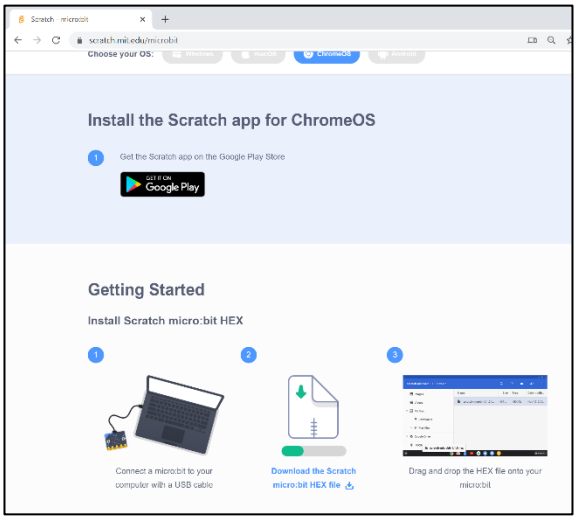
   ii. For ChromeOS and Android you will need to use the SCRATCH app to interface with the BBC micro:bit.

Download and install the SCRATCH hex file onto the BBC micro:bit.

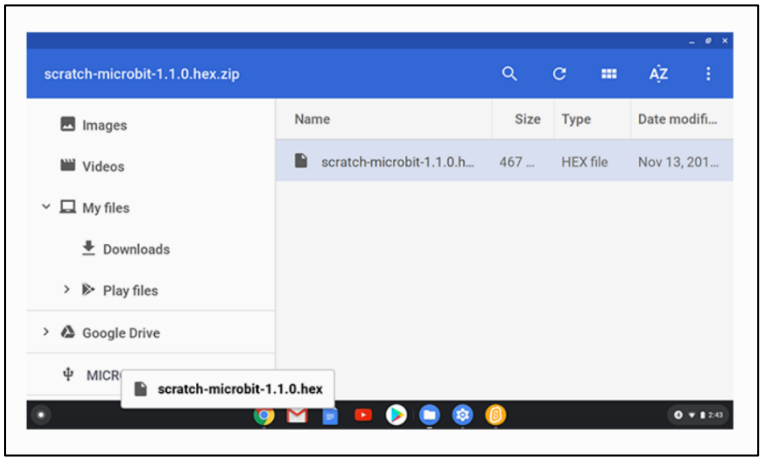
This step is the same regardless of operating system. This just needs to be completed once per BBC micro: bit device.

Note: If you are using an Android tablet or smartphone you must install the hex file via a computer through the micro USB attachment.

1. Download the hex file to your computer. You can do this directly from the SCRATCH BBC micro:bit website (https://scratch.mit.edu/microbit) or from the *Coding Whisperers* digital forum.



2. Connect the BBC micro:bit to the computer via the micro USB cable.
3. Drag and drop the hex code onto the BBC micro:bit.



Download and save the Flappy Bird scratch project file.

There are two versions of the Flabby Bird SCRATCH video game software on the *Coding Whisperers* digital forum:

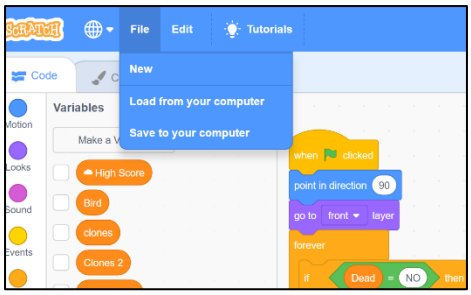
- Flappy Bird Scratch 3.0
- Flappy Bird Scratch 3.0 touch

The first (Flappy Bird Scratch 3.0) is the standard Flappy Bird code that is described in the Hack Flappy Bird investigation. The second (Flappy Bird Scratch 3.0 touch) is a modified version of the code to give students a head start if they want to use the capacitive touch sensor to modify the Flappy Bird video game. Instructions for this extension are provided with the code on the digital forum.

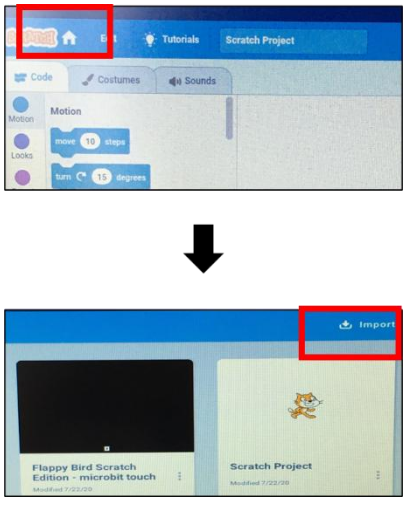
1. Download the Flappy Bird file and save it to some obvious place on each computer. (We suggest the desktop).

2. Students upload the Flappy Bird project file into SCRATCH to play and modify. The Flappy Bird project file can be uploaded into SCRATCH in either the browser version or the app version on of SCRATCH on any device. *Note: For ChromeOS and Android the BBC micro:bit (and any BBC micro:bit hacks) will only work in the app but the Flappy Bird project file will run in either version.*

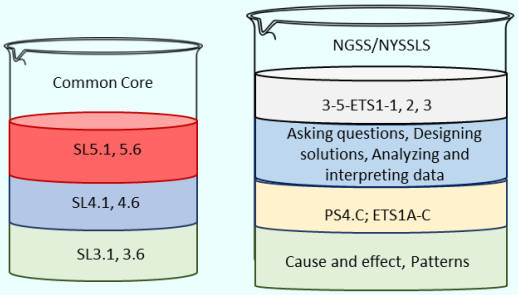
Browser version



App version



Investigation: Introduction to Scratch

Time	*Standards*	*Description*
		*In this investigation students gain experience with the level of detail and precision they must consider as they write computer programs.*

Related essential questions: Are computers smart? How do I write instructions for a computer?

Objective

This first investigation serves as a general introduction to coding and the SCRATCH software. It is likely that students will have a range of prior experience and coding can be quite intimidating for those who are new to it.

Materials

Computer or tablet—WiFi enabled

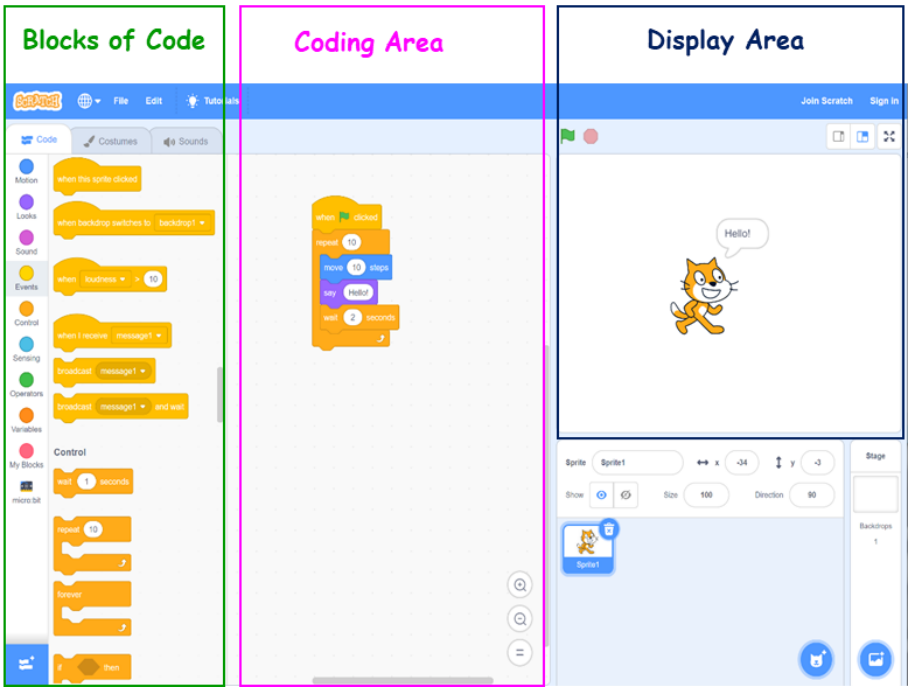
Detailed Description

Background

If SCRATCH is new to you, here are some more details.

a. SCRATCH is a "drag and drop" coding program developed at MIT (www.scratch.mit.edu). It takes all the nit-picky syntax requirements necessary for direct coding and allows users to write programs using "blocks" that perform certain functions.

b. These blocks are all located in the left column on the screen and can be dragged into the working area on the right. You can just snap the blocks of code together to create a program.



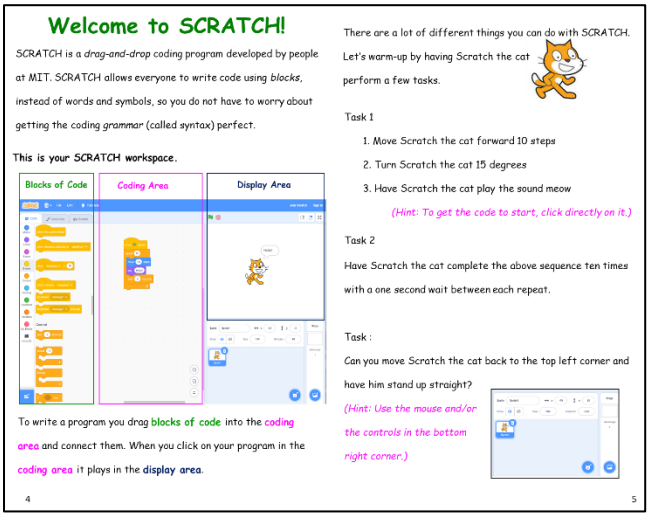
c. To start the program just click on anyone of the blocks in the sequence or on the green flag over the display area.

d. There is a whole community of coding and projects associated with SCRATCH. The website offers a plethora of opportunities for additional coding activities with your class.

Introduction

The goal of this introductory investigation is to make sure all your students understand the basics of coding with SRACTCH. If you haven't already shown the introductory SCRATCH video on the *Coding Whisperers* digital forum, show it now. This is a good opportunity to pull in questions about coding that arose during the introductory sessions.

To get started, a series of short tasks are outlined in the *My STEM Explorer Notes*™ notebooks on pages 4—6. There is also some space on page 7 for students to make notes about using the SCRATCH program.



Welcome to SCRATCH!

SCRATCH is a *drag-and-drop* coding program developed by people at MIT. SCRATCH allows everyone to write code using *blocks*, instead of words and symbols, so you do not have to worry about getting the coding *grammar* (called syntax) perfect.

This is your SCRATCH workspace.

Blocks of Code | Coding Area | Display Area

To write a program you drag blocks of code into the coding area and connect them. When you click on your program in the coding area it plays in the display area.

4

There are a lot of different things you can do with SCRATCH. Let's warm-up by having Scratch the cat perform a few tasks.

Task 1

1. Move Scratch the cat forward 10 steps
2. Turn Scratch the cat 15 degrees
3. Have Scratch the cat play the sound meow

(Hint: To get the code to start, click directly on it.)

Task 2

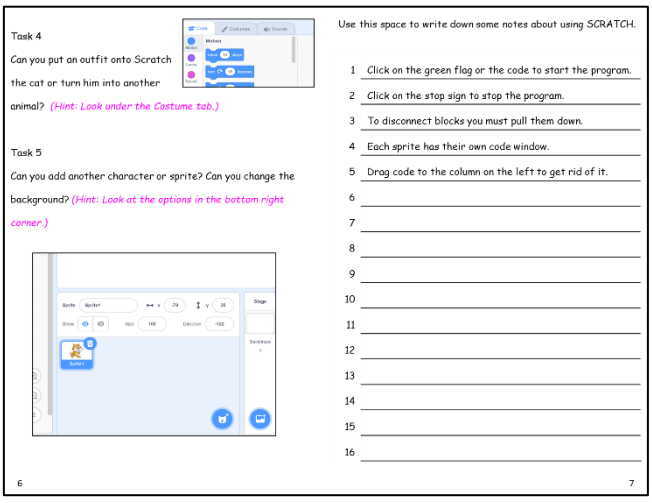
Have Scratch the cat complete the above sequence ten times with a one second wait between each repeat.

Task :

Can you move Scratch the cat back to the top left corner and have him stand up straight?

(Hint: Use the mouse and/or the controls in the bottom right corner.)

5



Task 4

Can you put an outfit onto Scratch the cat or turn him into another animal? *(Hint: Look under the Costume tab.)*

Task 5

Can you add another character or sprite? Can you change the background? *(Hint: Look at the options in the bottom right corner.)*

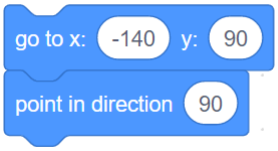
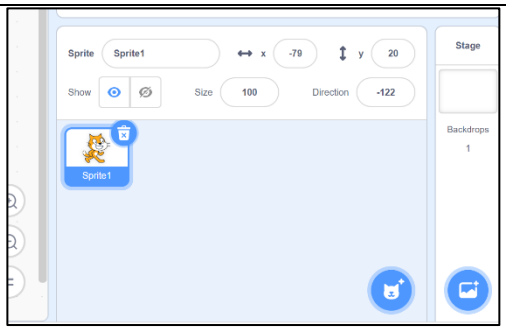
6

Use this space to write down some notes about using SCRATCH.

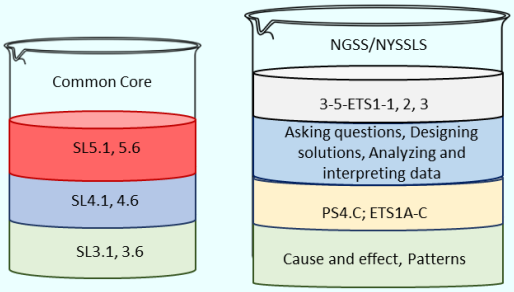
1 Click on the green flag or the code to start the program.
2 Click on the stop sign to stop the program.
3 To disconnect blocks you must pull them down.
4 Each sprite has their own code window.
5 Drag code to the column on the left to get rid of it.
6
7
8
9
10
11
12
13
14
15
16

7

While each task can be performed in a variety of ways, here are some examples of possible code.

Task	Code
I. Move Scratch the cat forward 10 steps II. Turn Scratch the cat 15 degrees III. Have Scratch the cat play the sound meow	
Have Scratch the cat complete the above sequence ten times with a one second wait between each repeat.	
Can you move Scratch the cat back to the top left corner and have him stand up straight?	
Can you put an outfit onto Scratch the cat or turn him into another animal? (Go to the costume tab on the left.)	
Can you add another character or sprite? Can you change the background? (Go to the workspace on the bottom right.)	

Investigation: Animation

Time	*Standards*	*Description*
		*In this investigation students have an opportunity to practice writing more complicated code. By giving students the freedom to choose most of the details around their animation they are able to explore coding in a way that is most interesting to them.*

Related essential question: How do I write instructions for a computer?

Objective

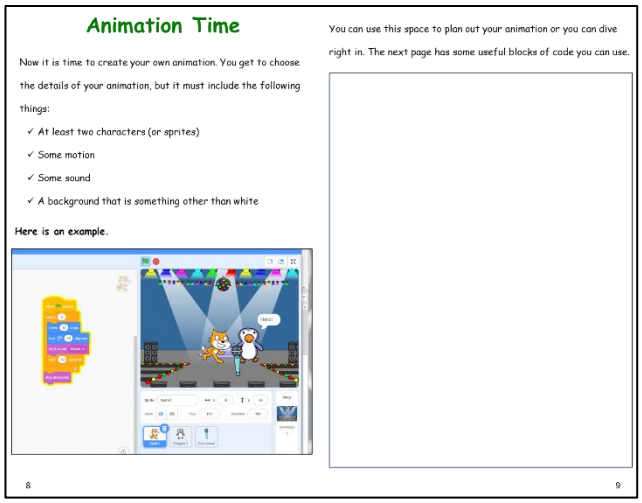
To give students a chance (and a reason) to explore and experiment with different SCRATCH blocks.

Materials

Computer or tablet—WiFi enabled.

Detailed Description

While the *My STEM Explorer Notes*™ notebook has some guidelines for the animation investigation, students generally need little prompting to start exploring. Additionally, there is a large community of coding and projects associated with SCRATCH and the website offers a plethora of opportunities for additional coding activities with your class.



Animation Time

Now it is time to create your own animation. You get to choose the details of your animation, but it must include the following things:

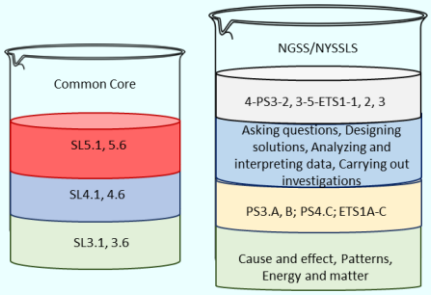
- ✓ At least two characters (or sprites)
- ✓ Some motion
- ✓ Some sound
- ✓ A background that is something other than white

Here is an example.

You can use this space to plan out your animation or you can dive right in. The next page has some useful blocks of code you can use.

8 9

Investigation: Introduction to the BBC micro:bit

Time	*Standards*	*Description*
		*In this investigation students incorporate external devices into their computer programming. They also start to explore the transfer of information and electricity between the computer and an external device (the BBC micro:bit).*

Related essential question: What are the different pieces needed to design a video game?

Objective

To give students an introduction to the BBC micro:bit.

Make sure never to connect the GND and 3V pin directly as you will produce a complete circuit with no resistance.

Materials

Computer or tablet—WiFi and Bluetooth enabled, BBC micro:bit, USB cable, alligator clipped wires.

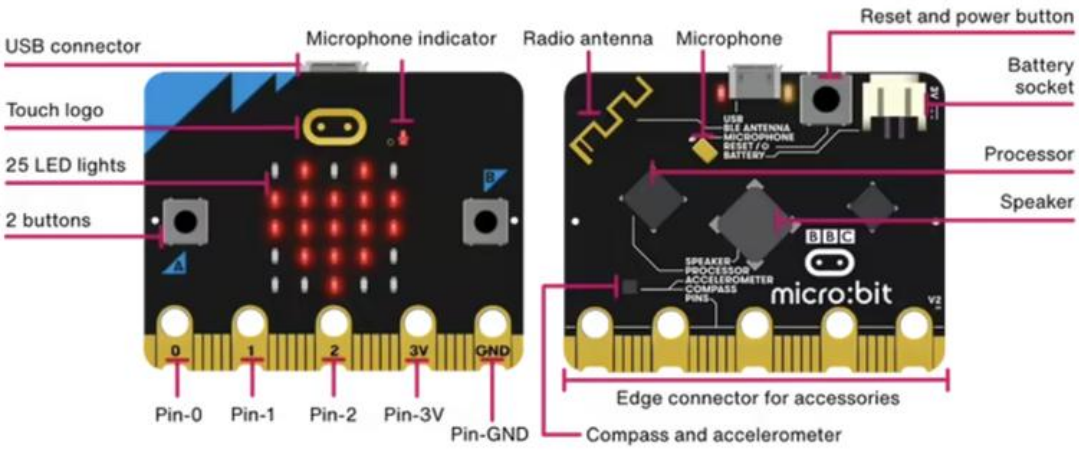
Detailed Description

Before you begin make sure the SCRATCH Link (Windows and macOS) or SCRATCH app (ChromeOS and Android) have been installed on your computer or tablet and that the hex code has been installed on the BBC micro:bit.

If you've decided to have your students participate in either installation process do that before beginning this investigation.

The BBC micro:bit is like a fancy controller that can be programed to do a lot of different things. It has buttons (like a video game controller), a motion sensor (like airplane controls), and can act as a communication bridge between the SCRATCH code and the outside world.

BBC micro:bit layout



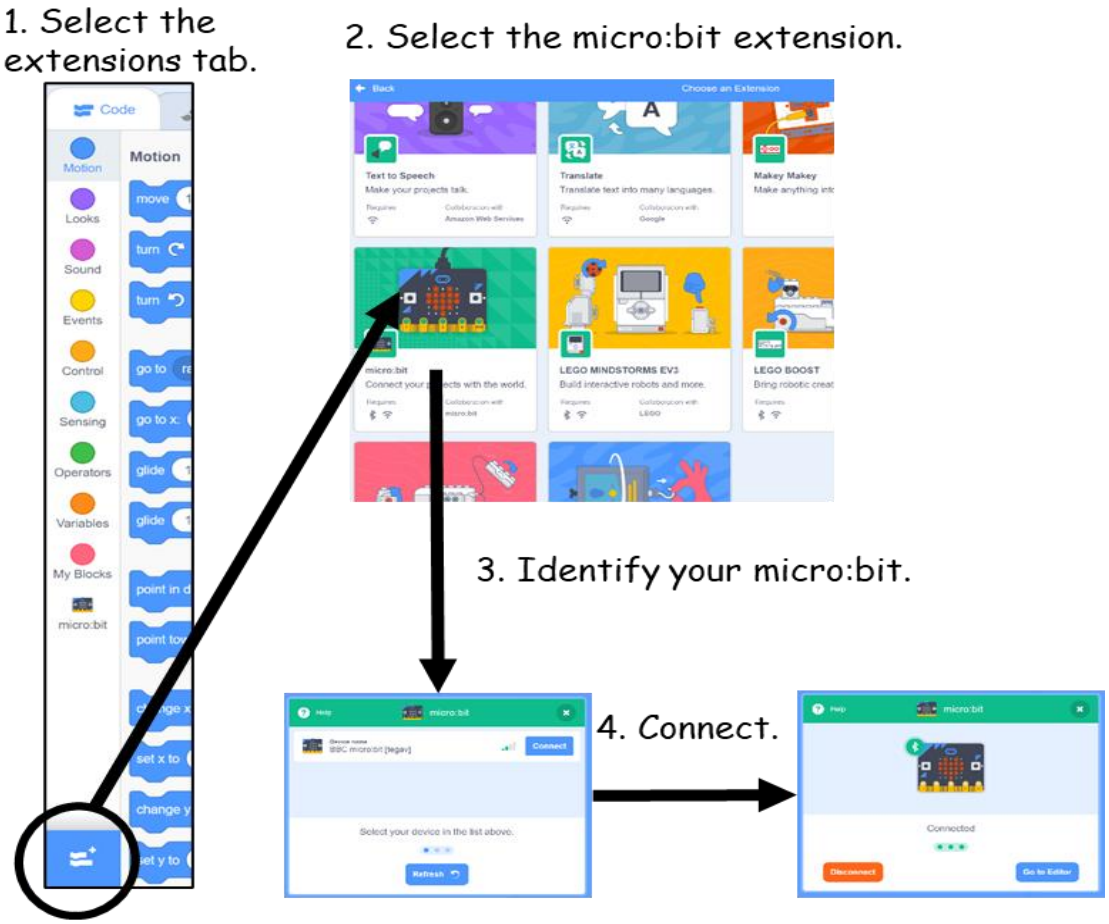
Make sure never to connect the GND and 3V pin directly.

Connecting the BBC micro:bit

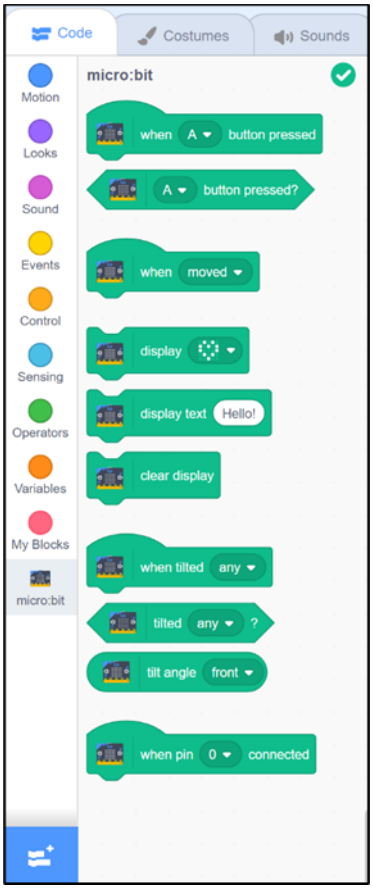
- Make sure the WiFi and Bluetooth functions are on.
- Connect the BBC micro:bit to the computer via the USB port. This provides the power for the BBC micro:bit. If you are using a tablet device you can hook up an external battery to the BBC micro:bit via the battery socket.

- Open up SCRATCH (Windows or macOS) or the SCRATCH app (ChromeOS or Android).
- Connect SCRATCH and the BBC micro:bit via the following steps:

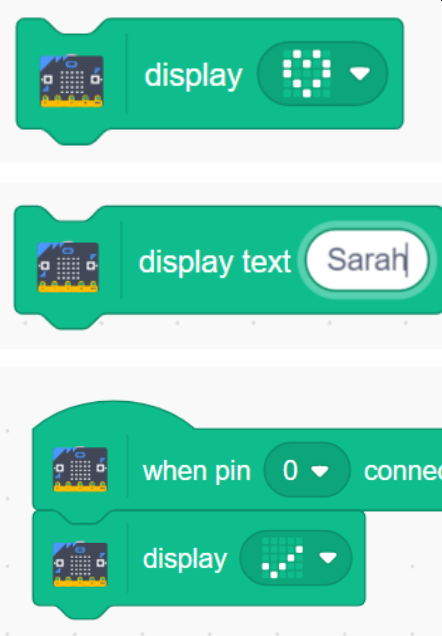
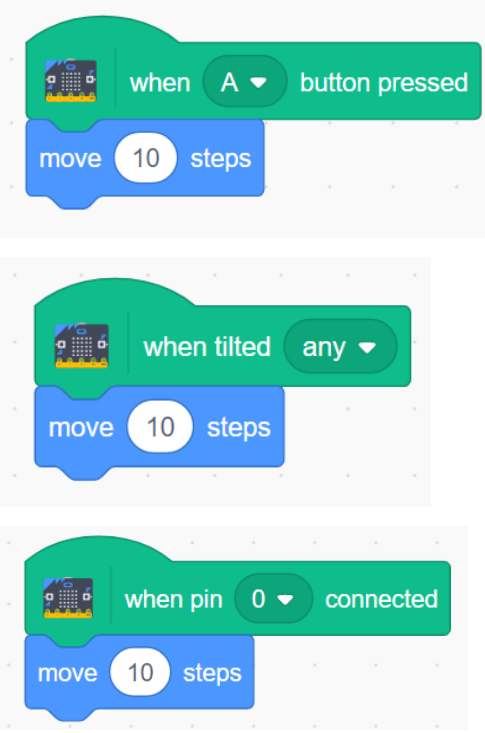
- Select the extensions tab.
- Select the micro:bit extension.
- Identify your micro:bit. This may get a bit confusing if you have a number of micro:bit in close proximity. The “name” of the micro:bit should scroll across the display as it is searching for a Bluetooth connection.
- Connect.

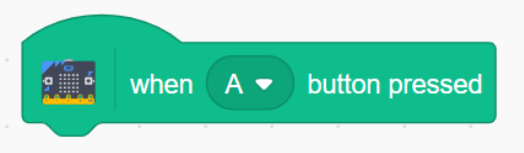
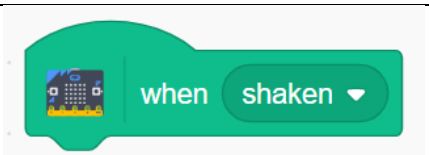
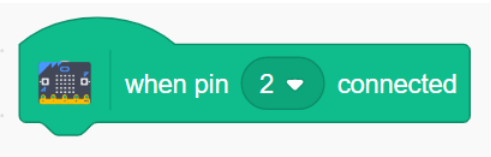
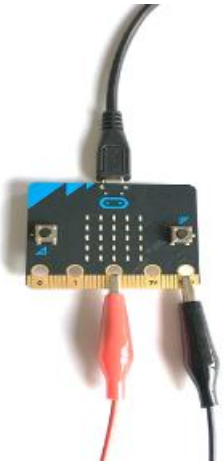


Upon connecting to the BBC micro:bit an additional series of blocks should show up in the left-hand column. The green checkmark in the right of this column indicates that the BBC micro:bit is successfully connected.

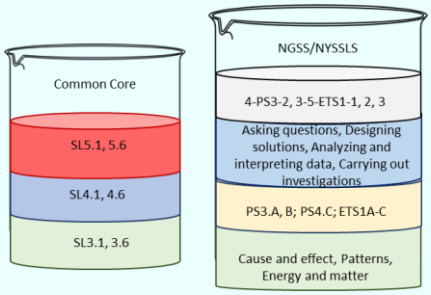


As with the general introduction to SCRATCH, a few specific tasks have been outlined to help students familiarize themselves with the BBC micro:bit blocks.

Task	Code
Can you… • light up a heart on the LED display? • have the LED display show your name? • change the LED display by connecting a pin?	
Can you… • control Scratch the cat by pressing a button on the micro:bit? • control Scratch the cat by moving the micro:bit? • control Scratch the cat with a command started by a pin?	

Task	Code
Can you use either the A or B button in your animation?	
Can you use the jump or shake function in your animation?	
Can you use pin 2 in your animation?	 

Investigation: Can You Hack Flappy Bird?

Time	*Standards*	*Description*
		*In this investigation students build upon their learning about incorporating external devices and start to explore the details and complexity required to design a video game.*

Related essential questions: What are the different pieces needed to design a video game? How do I design a video game?

Objective

To give students some experience working with and manipulating more complicated code (many sets of code blocks, a number of different sprites, etc.)

As always, make sure never to connect the GND and 3V pin directly as you will produce a complete circuit with no resistance.

Materials

Computer or tablet—WiFi and Bluetooth enabled, BBC micro:bit, USB cable, alligator clipped wires.

Detailed Description

<u>Quick Flappy Bird Introduction</u>

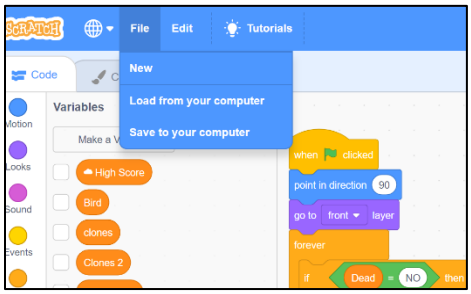
While most students have heard of the video game flappy bird, most adults have not. The goal is to make a bird fly through the opening in green pipes. It is a simple premise that is actually much harder to do than you would expect. The main video game runs on either the browser or app version of SCRATCH. The game, and game code, was written specifically to give students a

chance to “hack” it, i.e. change around the code. Before we go into how to “hack” Flappy Bird, here is a quick overview of how the program works.

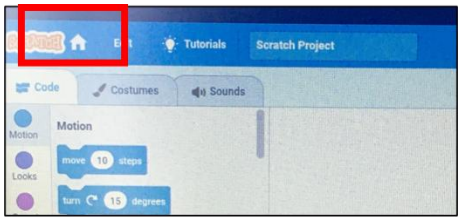
I. Make sure the Flappy Bird project file is saved on your computer. We recommend the desktop.

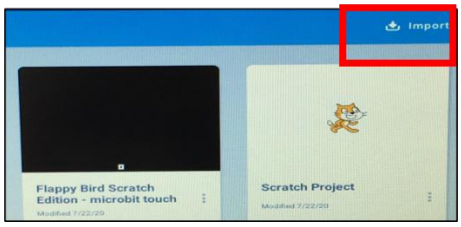
II. Go to SCRATCH and select File/Load from computer or Home/Import from the App. Note: Once Flappy Bird is imported into the app is does not need to be imported again.

Browser version

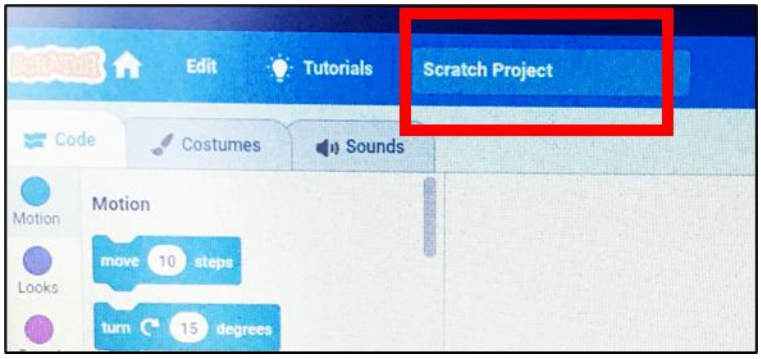


App version



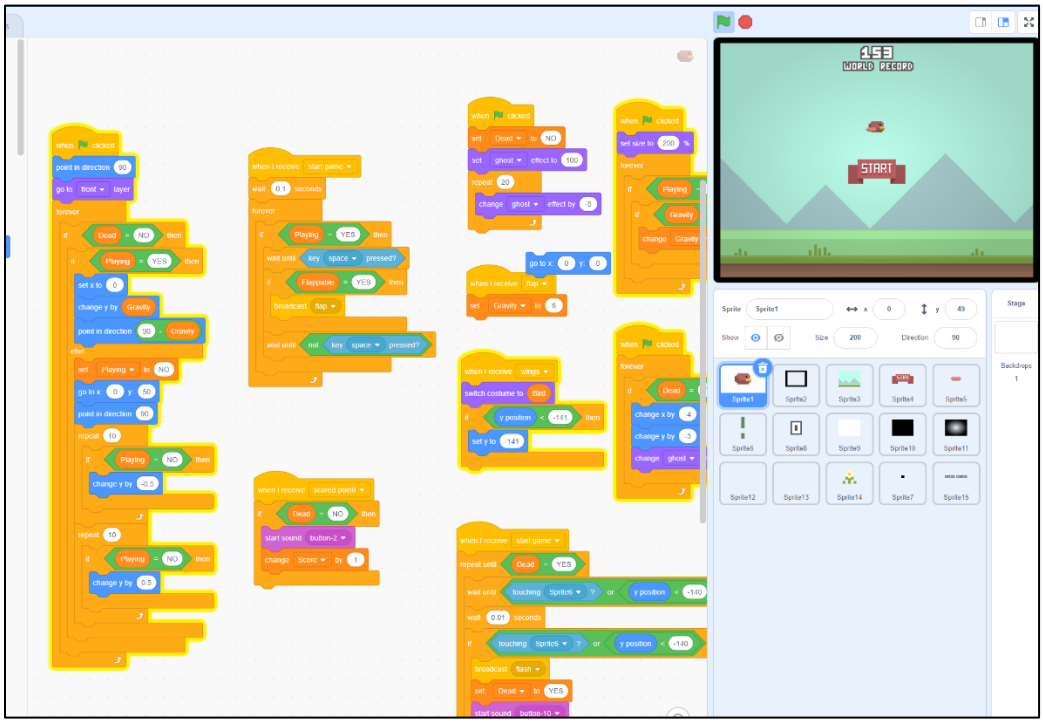


When working in the app we suggest students change the name of the file immediately to avoid saving over the original file.



III. Hit the green flag on the top right of the video screen to “activate” the game.

IV. Hit the space bar to start the game and to make Flappy Bird fly. See how many pipes you can get Flappy Bird to fly through (my best score is 6).



In this investigation students can change several parts of the Flappy Bird code. At any point, if the program is changed so much that it stops working you can always reload the project from your computer or from the *Coding Whisperers* digital forum.

While students rarely need much encouragement to come up with different ways to hack the code, a couple of suggestions to get their minds going are listed on pages 16 and 17 of their *My STEM Explorer Notes*™ notebooks.

Can You Hack Flappy Bird?

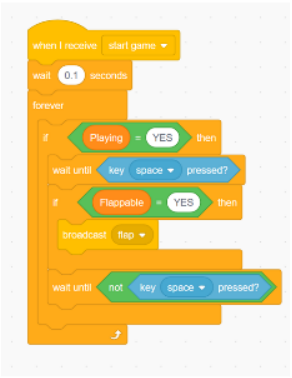
To load Flappy Bird go to ***File/Load from your computer*** and select the Flappy Bird Scratch Edition.sb3. It should be on your desktop.

By changing the code you can change the way the game is played (and scored). Can you make the following changes?

1. Ensure that you get a new high score as soon as you clear the first obstacle.
2. Make it easier for the Flappy Bird sprite to get through the obstacles.
3. Change how the Flappy Bird sprite flaps:
   - Use a different keyboard key
   - Use a motion on the BBC micro:bit
   - Use a button on the BBC micro:bit
4. Can you add something to the Flappy Bird code that shows your score on the BBC micro:bit display?

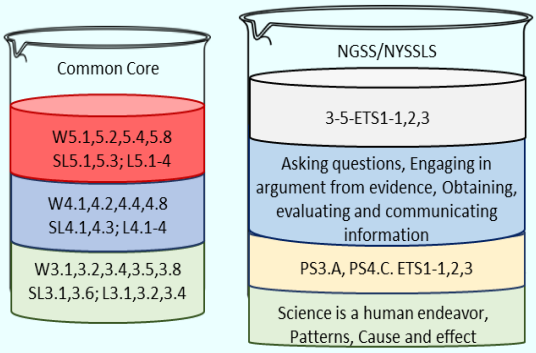
16

Here are some pieces of the Flappy Bird code that might help you with your modifications.



17

Design Challenge

Time	*Standards*	*Description*
		*The design challenge provides students with the opportunity to apply the concepts they have learned and practiced during the investigations to a broader and deeper project. To complete this challenge, they must not only* ***know*** *the specific standards but also understand* ***why*** *that knowledge is useful and how they can apply it to a new problem or application. Students should work in teams of three or four to complete one of the following projects and, if time allows, present their findings to the class.*

Related essential questions: How do I design a video game?

Design Your Own Video Game Design Challenge

The Design Your Own Video Game design challenge is a chance for students to apply what they have been learning individually to a group project. One of the advantages of coding is that it lends itself intellectual collaboration even if the team members are physically distant. It is also a great opportunity for cross-curricular integration specifically with art, ELA, and math. Students have space in their *My STEM Explorer Notes*™ notebooks to think about and plan out their video game.

Each video game should include the following things:

- A video game (A)
- Clear instructions on how to play the game. (B)*
- A short description by each individual describing what they did and how they did it. (C)*

*B and C can be either written or completed as a short video.

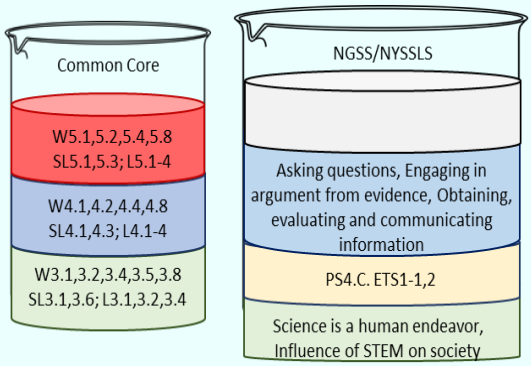
Rubric:

Score	A	B	C	Teamwork
3	Functioning video game	Complete, well-articulated instructions	Complete description of what was done and how it was done (their individual part only)	In addition to below, members worked to encourage and teach each other
2	Good attempt at a video game but it doesn't work as intended	Complete but hard to understand instructions	Complete description of what was done but no description of how it was done (their individual part only)	Team functioned well with all members contributing
1	Video game outline but very little coded	Incomplete instructions	Incomplete description of what was done (their individual part only)	Team functioned well most of the time but some members were more engaged than others
0	No video game	No instructions	No description	Team required a lot of adult intervention to ensure all members contributed/ were included

Summative Challenge

Time

Standards



Description

Summative challenges provide students with the opportunity to apply the concepts they have learned and practiced during the investigations to a broader and deeper project. To complete this challenge, they must not only ***know*** *the specific standards but also understand* ***why*** *that knowledge is useful and how they can apply it to a new problem or application. Students should work in teams of three or four to complete one of the following projects and, if time allows, present their findings to the class.*

Computing Solutions Summative Challenge

Two common themes in this unit are a) the recognition that coding and computer programming are a part of our everyday life and b) that coding and computer programming provide a great opportunity to help solve problems in the community. The Computing Solutions summative challenges gives students an opportunity to reflect and build upon these two themes. This is also a good opportunity to pull in student questions that have arise throughout the unit and have them incorporate them into their projects.

- Create a poster or brochure to educate the public about the following things:
    - How do programming and microprocessors work? (A)
    - How are they used in everyday life (give at least 5 examples)? (B)
    - How can improved programming technology help the disabled or elderly? Identify at least two devices that could be improved or use this technology to help someone move around their homes, prepare food, get dressed, etc. and explain why they are important. (C)

Rubric:

Score	A	B	C	Teamwork
3	Complete information and well written	Complete answer and well written	Complete answer and well communicated	In addition to below members worked to encourage and teach each other
2	Complete information (includes all components) but poorly written	A single example with reasonable justification	Adequate examples but no justification	Team functioned well with all members contributing
1	Partial information, describes some components of batteries but not all	A single example with no explanation	Incomplete response with no justification	Team functioned well most of the time but some members were more engaged than others
0	No information	No or incomplete examples	No answer	Required a lot of adult intervention to ensure all members contributed

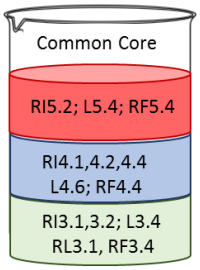
- Students should design a new toy or game that can be used both with programming and without. Students can create a poster to describe their toy or game. The poster should include:
    - A picture of the game or toy that highlights the different features/ways the toy or game is used with and without programming (A)
    - A written description of how the toy works both with programming and without. (B)
    - A "product review" from each team member describing why they like the toy and why other people should try it out. (C)

Rubric:

Score	A	B	C	Teamwork
3	Complete picture and description	Complete answer, well written	Complete set of reviews well written	In addition to below members worked to encourage and teach each other
2	Complete picture but without the differences highlighted	Complete description or instructions but poorly written	Product review written by everyone but are not complete	Team functioned well with all members contributing
1	Incomplete picture	Description or instructions describe only one mode of use	Only some team members complete product reviews	Team functioned well most of the time but some members were more engaged than others
0	No picture	No description or set of instructions	No product reviews	Required a lot of adult intervention to ensure all members contributed

Ongoing Cross-Curricular Activities

The cross-curricular activities provided in the next few pages are just some suggestions on how you can deepen and continue the discussion on water resources and adaptations.

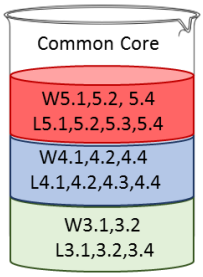


Reading Comprehension Formative Assessment Suggestions

The topic introduction, personal biographies, and company information can all be used as informational text reading. They can also be used to reinforce connections between the scientific concepts students are learning about and the bigger picture thinking encouraged by the phenomena, enduring understanding and essential questions. Text and questions can be found in the *My STEM Stories*™ notebooks.

1. Have the students use each of the vocabulary words in a new sentence.
2. Have the students read the introduction in the *My STEM Stories*™ notebook and then answer the following:
    a. How is coding like a spoken language?
    b. Describe three ways you interact with coding or computer programming every day?
3. Using the entrepreneur and innovator information in the *My STEM Stories*™ notebook, answer the following questions:
    a. How are innovators and entrepreneurs using coding to help people? Describe a problem you see in your home, school, or neighborhood, and discuss how coding might be used to help with a solution

Written Reflections or Discussions



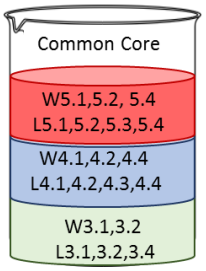
Written reflections and discussions are a great way for students to revisit some of bigger thinking questions in the unit while also working on their writing skills. These open-ended questions encourage students to draw from what they have learnt and apply it to the larger community. This is also a good opportunity to revisit questions that students have raised throughout the unit.

1. How can coding be used to increase help us accomplish more in a shorter amount of time as well as to give us more ways to have fun?
2. How can I use coding to solve a problem in my life or community?
3. Given Earth's limited resources, how can we use technology to accomplish more?

Creative Writing

- Set aside 15 minutes each week to work in the *Book of Ideas* (if purchased)
- Have the students write a story using all the vocabulary words

Additional Writing Prompts



1. Write three facts you learned today.
2. What surprised you the most about what you learned today?
3. How does this technology connect with things you or your family do?
4. Look around your home and/or school. Identify at least 5 devices that have microcomputers in them.
5. Identify and describe three reasons/ways this technology is good for humans.
6. Look around your home and/or school. Identify at least one device that could be changed so it could use a microcomputer. Why would that device be better if it used a microcomputer?

Social Studies

- Have students complete one of the suggested timeline activities (see Appendix for suggestions).

Math Word Problems

- Students can find 10 objects with computers or microprocessor in or around their house and categorizes them as for work, for pleasure (e.g. the TV) or for everyday life (e.g. a car or cell phone). Create a line plot of the results and then share with the class to create a bar graph.

- If the class has 25 minutes to play Flappy Bird, and there are 5 computer stations, how many total minutes of video game playing do they have? If the class has 125 students how many students are assigned to each station? If the total playing time is divided evenly between the whole class, how much time does each student have to play the game?

Pacing Guide Resources

This section contains a sample pacing guide for you to use as a resource when planning out how to teach this unit in your class, including lesson sequencing, cross-curricular integration, external resources, and quick assessments to monitor ongoing student understanding. The last page in this section is intentionally left blank for you to develop the outline for your own lesson plan. **More sheets are provided at the end of the Appendix to support the creation of your detailed lesson plans**. Feel free to make modifications to this curriculum in response to your students' interests and needs. If you need additional support or guidance in making modifications, while making sure you are still addressing all the concepts covered in the standards, please feel free to reach out to the CreositySpace curriculum development team by emailing Kath at Kath@CreositySpace.com.

Guiding notes for teachers as you are developing your lesson plans and pacing guide

- Review Learning Progression, Entrepreneur stories, and Essential Questions.
- Use provided lesson planning sheets to outline lesson flow and highlight connections to learning objectives you have in other subjects, especially math, ELA, and social studies.
- Determine a specific introduction strategy for your class.
- Review how-to videos and safety section for hands-on investigations. Determine any **additional safety precautions** you should highlight to keep ALL students safe during the investigations.
- Review the suggested videos and discussion prompts online.
- Verify access and compatibility of your school's computers and internet access with SCRATCH (see introduction section for details). Contact us (Kath@Creosityspace.com) ASAP if any issues arise.
- Review Ongoing Cross-Curricular Activities (Lesson section) and Additional Resources for implementation throughout the unit and year in general.
- Each week has several suggested Exit Tickets that can be used in various ways (e.g., journal entries, writing prompts, recap videos, small group discussions, etc.) as an ongoing formative assessment of student understanding.
- Review **Teacher Support Documents** in the Appendix for various tips on things such as STEM integration, getting back on-track after an extended student-directed discussion, lesson scaffolding, etc.
- **A composition journal that students can use as a STEM notebook (to support brainstorming, inquiry, research, writing, etc.) is strongly suggested.**

Supplemental Program

This five-session guide is intended as an example for folks using *Coding Whisperers* as a ***supplemental curriculum*** with a focus on **physical computing**. It is expected that each "session" contains at least two 40-minute class periods, although many can be extended far beyond that should you and your students wish to explore any one section in more detail. All activities described can be a part of your science class, however several lessons are also suitable for instruction during ELA and, to a lesser extent, social studies, art or math classes. Lessons that are suitable for an ELA-, social studies-, art-, or math-focused instruction block are presented in purple text.

. ***Note:*** *All links were confirmed as working at the time this Educator Guide was created. If you find a link that doesn't work, please let us know.*

Session	Preparation and Lessons

1
Introduction to Coding
	Prep Activities
	• Read introductory section
	• Review Ongoing Cross-Curricular Activities for implementation throughout the unit and year in general.
	
	Intended Activities
	• Introduce the topic area
	  ○ Discuss introductory **phenomenon:** *What would you need to make your own video game?*
	  ○ Complete introductory Challenge – *Robot Chef*
	  ○ Show an introductory video:
	  ▪ **DK Coding for Kids video 1 or 2**
	
	Optional Activities
	• Distribute My STEM Stories™ notebooks
	  ○ Read and discuss introduction
	  ○ Read and discuss entrepreneur stories.
	
	Exit Tickets
	What does programming allow you to do?
	Are computers smart? Why or why not?

2
Introduction to SCRATCH
	Prep Activities
	• Review the company video and activity tutorials online.
	• Make sure SCRATCH isn't blocked by your school's firewall.
	
	Intended Activities
	• If not already done, distribute My STEM Stories™ notebooks
	  ○ Show the *Coding Whisperers* lesson video.
	  ○ Discuss key vocabulary with the students.
	• Complete the **Introduction to SCRATCH** and the **Animation** investigations.
	  ○ Show an introductory video:
	  ▪ **DK Coding for Kids video 3**
	
	Optional Activities
	• Do a Technology Historical Timeline activity.
	• **Writing Prompt:** What kind of projects can you create with different coding?
	
	Exit Ticket
	Why is it helpful to plan out your program before you start coding?

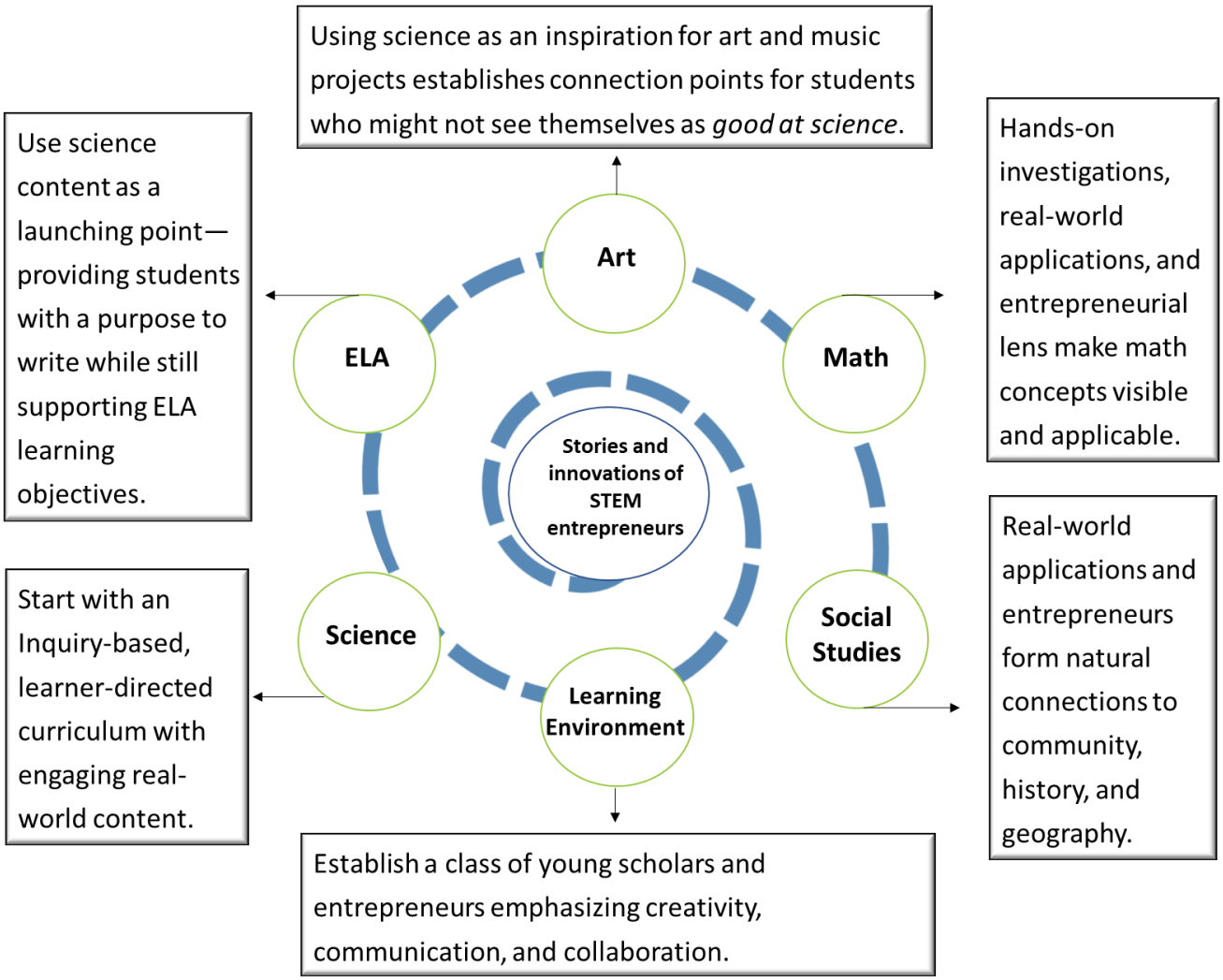
3 Introduction to the BBC micro:bit	Intended Activities • Complete the **Introduction to the BBC micro:bit and Can You Hack Flappy Bird** investigations ? Optional Activities • **Writing prompt:** If you could turn something into a computer what would it be and why? • Introduce the *Book of Ideas* (if ordered) https://www.creosityspace.com/young-entrepreneur-tools.html Exit Ticket Describe physical computing in your own words.

4
Design
Challenges
	Intended Activities
	• Complete the design challenge **Design You Own Video Game**. This will take at minimum five class periods (game plus written or recorded instructions and discussion). If you have more time encourage students to include the following variations:
	  ○ Multiple characters and scenery pieces.
	  ○ Externally recorded or imported sounds or music.
	  ○ Original artwork.
	
	Optional Activities
	• **Writing prompt:** Complete a creative writing activity.
	
	Exit Tickets
	What is an LED?
	Name three things you could program and how you know they are programmable.

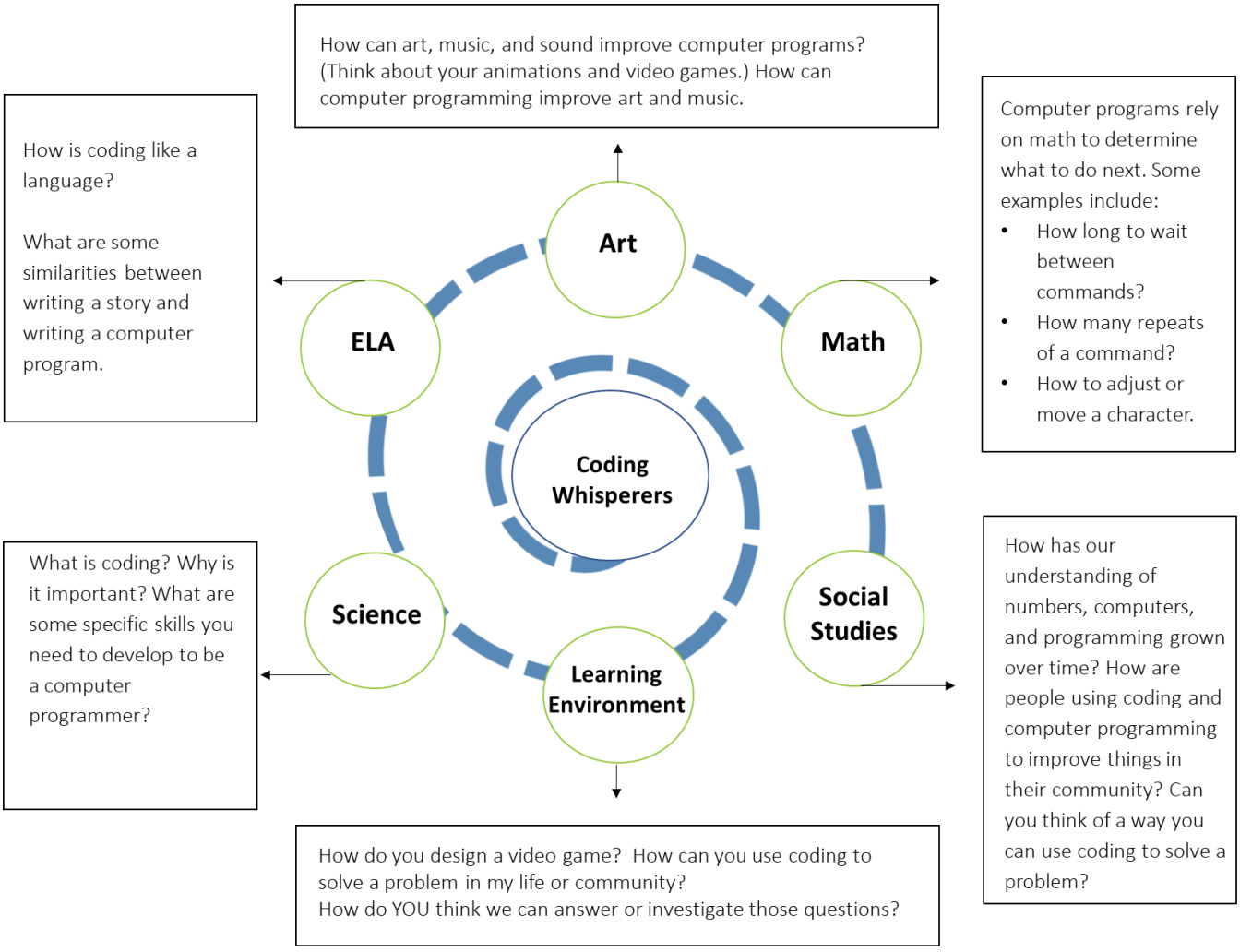
5 Ongoing Challenges and Cross-Curricular Activities	• **Writing prompt:** Complete a writing reflection piece. • Incorporate math and writing prompts from the Ongoing Cross-Curricular Activities section into your ELA and math classes. • Begin a summative challenge.

Cross-Curricular Integration

Cross-curricular integration is a great way to save time and increase engagement. In the next few pages we describe some examples of how this can be achieved. **The key to success: Students must see how what they are learning connects to their interests.**



Note: An example of using this spiral as a planning tool is provided on the next page.



The unit webpage contains other examples of how classroom teachers have integrated this unit across all of their content areas.

The blank spirals are also a great tool to give your students an opportunity to show you what they know from the unit.

Note: A blank spiral template that you can use to plan out a cross-curricular integration strategy for your classroom is provided in the Appendix.

Blank Pacing Guide

Use this page to summarize your plans for using *Water Watchers*. Pages for weekly lesson planning are provided in the Appendix.

Introduction:
Lesson Flow:
Summative Assessment:

Education Standards

Don't see the standards for your school district? Contact us at kath@creosityspace.com and we will determine the appropriate standards alignment for your district.

**Next Generation Science Standards (NGSS) is a registered trademark of WestEd. Neither WestEd nor the lead states and partners that developed the standards were involved in the production of, and does not endorse, this product.*

Next Generation Science Standards/NY State Science Learning Standards 3—5

Performance Expectations

4-PS3-2. Make observations to provide evidence that energy can be transferred from place to place by sound, light, heat, and electric currents.
4-PS4-3. Generate and compare multiple solutions that use patterns to transfer information.

3-5-ETS1-1. Define a simple design problem reflecting a need or a want that includes specified criteria for success and constraints on materials, time, or cost.
3-5-ETS1-2. Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.
3-5-ETS1-3. Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved.

Science and Engineering Practices

Asking questions / defining problems
Hands-on coding investigations are all about asking questions and figuring out how to achieve a task.
Reflections on using coding to support a community need also has students asking questions about the world around them.

Using math & computational thinking, analyzing and interpreting data
Coding investigations require math and computational thinking. They also require students to evaluate how their code has (or has not) worked and what changes they should make.

Constructing explanations/designing solutions
Design and summative challenges, innovation prompts, and overall reflections have students constructing explanations and designing solutions.

Obtaining, evaluating, and communicating information
Requirements for students to explain their codes and video games have students working on effective ways to communicate information.

Connections to Nature of Science

Scientific investigations use a variety of methods
Entrepreneur stories and hands-on investigations illustrate how scientific investigations are conducted and how that information is put to use.

Disciplinary Core Ideas

PS3.A Definitions of energy
PS4.C Information technologies and instrumentation
Coding puts into practice the theory of electricity and information technologies.

ETS1.A: Defining and Delimiting Engineering Problems
ETS1.B: Developing Possible Solutions
ETS1.C: Optimizing the Design Solution
Science and technology based writing prompts, challenge questions and group activities support the three phases of Engineering Design.
The *Book of Ideas*, along with the Design Challenge lesson plan take students through the full cycle of concept to design.

Cross Cutting Concepts

Patterns; Cause and effect; Energy and matter: Flows, cycles, and conservation
Elements of coding dive into dissecting how and why something happens and then to replicate it in code.

Connections to Nature of Science

Science is a way of knowing; Science addresses questions about the natural and material world
Videos and introduction text give support these connections.

Science is a human endeavor
Entrepreneur story and historical timeline highlight the human aspect of science and engineering.

Connections to Engineering, Technology, and Applications of Science

Interdependence of Science, Engineering, and Technology; Influence of Engineering, Technology and Science on Society and the Natural World
Introduction text, historical timeline, entrepreneur story and activities highlight above interactions and interdependencies.

Connections to Common Core State Standards
See previous Common Core Standards section for ELA and Math standards addressed by these activities.

Common Core ELA Standards

Grade 3

Reading Informational Text:

CCSS.ELA-LITERACY.RI.3.1 Ask and answer questions to demonstrate understanding of a text, referring explicitly to the text as the basis for the answers.
CCSS.ELA-LITERACY.RI.3.2 Determine the main idea of a text; recount the key details and explain how they support the main idea.
CCSS.ELA-LITERACY.RI.3.4 Determine the meaning of general academic and domain-specific words and phrases in a text relevant to a *grade 3 topic or subject area*.

Writing:

CCSS.ELA-Literacy.W.3.2 Write informative/explanatory texts to examine a topic and convey ideas and information clearly.
CCSS.ELA-Literacy.W.3.2.a Introduce a topic and group related information together; include illustrations when useful to aiding comprehension.
CCSS.ELA-Literacy.W.3.2.b Develop the topic with facts, definitions, and details.
CCSS.ELA-Literacy.W.3.2.c Use linking words and phrases (e.g., *also*, *another*, *and*, *more*, *but*) to connect ideas within categories of information.
CCSS.ELA-Literacy.W.3.2.d Provide a concluding statement or section.
CCSS.ELA-Literacy.W.3.1 Write opinion pieces on topics or texts, supporting a point of view with reasons.
CCSS.ELA-Literacy.W.3.1.a Introduce the topic or text they are writing about, state an opinion, and create an organizational structure that lists reasons.
CCSS.ELA-Literacy.W.3.1.b Provide reasons that support the opinion.
CCSS.ELA-Literacy.W.3.1.c Use linking words and phrases (e.g., *because*, *therefore*, *since*, *for example*) to connect opinion and reasons.
CCSS.ELA-Literacy.W.3.1.d Provide a concluding statement or section.
CCSS.ELA-Literacy.W.3.4 With guidance and support from adults, produce writing in which the development and organization are appropriate to task and purpose. (Grade-specific expectations for writing types are defined in standards 1-3 above.)
CCSS.ELA-Literacy.W.3.5 With guidance and support from peers and adults, develop and strengthen writing as needed by planning, revising, and editing.
CCSS.ELA-Literacy.W.3.6 With guidance and support from adults, use technology to produce and publish writing (using keyboarding skills) as well as to interact and collaborate with others.
CCSS.ELA-Literacy.W.3.7 Conduct short research projects that build knowledge about a topic.
CCSS.ELA-Literacy.W.3.8 Recall information from experiences or gather information from print and digital sources; take brief notes on sources and sort evidence into provided categories.

Speaking & Listening:

CCSS.ELA-LITERACY.SL.3.1 Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on *grade 3 topics and texts*, building on others' ideas and expressing their own clearly.
CCSS.ELA-LITERACY.SL.3.1.A Come to discussions prepared, having read or studied required material; explicitly draw on that preparation and other information known about the topic to explore ideas under discussion.
CCSS.ELA-LITERACY.SL.3.1.B Follow agreed-upon rules for discussions (e.g., gaining the floor in respectful ways, listening to others with care, speaking one at a time about the topics and texts under discussion).
CCSS.ELA-Literacy.SL.3.1.c Ask questions to check understanding of information presented, stay on topic, and link their comments to the remarks of others.
CCSS.ELA-Literacy.SL.3.1.d Explain their own ideas and understanding in light of the discussion.
CCSS.ELA-Literacy.SL.3.3 Ask and answer questions about information from a speaker, offering appropriate elaboration and detail.
CCSS.ELA-Literacy.SL.3.4 Report on a topic or text, tell a story, or recount an experience with appropriate facts and relevant, descriptive details, speaking clearly at an understandable pace.
CCSS.ELA-Literacy.SL.3.6 Speak in complete sentences when appropriate to task and situation in order to provide requested detail or clarification.

Language:

CCSS.ELA-LITERACY.L.3.1 Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.
CCSS.ELA-LITERACY.L.3.1.A Explain the function of nouns, pronouns, verbs, adjectives, and adverbs in general and their functions in particular sentences.
CCSS.ELA-LITERACY.L.3.2 Demonstrate command of the conventions of standard English capitalization, punctuation, and spelling when writing.
CCSS.ELA-LITERACY.L.3.4 Determine or clarify the meaning of unknown and multiple-meaning word and phrases based on grade 3 reading and content, choosing flexibly from a range of strategies.

Grade 4

Reading Informational Text:

CCSS.ELA-LITERACY.RI.4.1 Refer to details and examples in a text when explaining what the text says explicitly and when drawing inferences from the text.
CCSS.ELA-LITERACY.RI.4.2 Determine the main idea of a text and explain how it is supported by key details; summarize the text.
CCSS.ELA-LITERACY.RI.4.4 Determine the meaning of general academic and domain-specific words or phrases in a text relevant to a *grade 4 topic or subject area*.
CCSS.ELA-LITERACY.RI.4.5 Describe the overall structure (e.g., chronology, comparison, cause/effect, problem/solution) of events, ideas, concepts, or information in a text or part of a text.
CCSS.ELA-LITERACY.RI.4.7 Interpret information presented visually, orally, or quantitatively (e.g., in charts, graphs, diagrams, time lines, animations, or interactive elements on Web pages) and explain how the information contributes to an understanding of the text in which it appears.

Writing:

CCSS.ELA-LITERACY.W.4.1 Write opinion pieces on topics or texts, supporting a point of view with reasons and information.
CCSS.ELA-LITERACY.W.4.1.A Introduce a topic or text clearly, state an opinion, and create an organizational structure in which related ideas are grouped to support the writer's purpose.
CCSS.ELA-LITERACY.W.4.1.B Provide reasons that are supported by facts and details.
CCSS.ELA-LITERACY.W.4.1.C Link opinion and reasons using words and phrases
CCSS.ELA-LITERACY.W.4.1.D Provide a concluding statement or section related to the opinion presented.
CCSS.ELA-LITERACY.W.4.2 Write informative/explanatory texts to examine a topic and convey ideas and information clearly.
CCSS.ELA-LITERACY.W.4.2.A Introduce a topic clearly and group related information in paragraphs and sections; include formatting (e.g., headings), illustrations, and multimedia when useful to aiding comprehension.
CCSS.ELA-LITERACY.W.4.2.B Develop the topic with facts, definitions, concrete details, quotations, or other information and examples related to the topic.
CCSS.ELA-LITERACY.W.4.2.C Link ideas within categories of information using words and phrases.
CCSS.ELA-LITERACY.W.4.2.D Use precise language and domain-specific vocabulary to inform about or explain the topic.
CCSS.ELA-LITERACY.W.4.2.E Provide a concluding statement or section related to the information or explanation presented.
CCSS.ELA-LITERACY.W.4.4 Produce clear and coherent writing in which the development and organization are appropriate to task, purpose, and audience.
CCSS.ELA-LITERACY.W.4.8 Recall relevant information from experiences or gather relevant information from print and digital sources; take notes and categorize information, and provide a list of sources.

Speaking & Listening:

CCSS.ELA-LITERACY.SL.4.1 Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on *grade 4 topics and texts*, building on others' ideas and expressing their own clearly.
CCSS.ELA-LITERACY.SL.4.1.A Come to discussions prepared, having read or studied required material; explicitly draw on that preparation and other information known about the topic to explore ideas under discussion.
CCSS.ELA-LITERACY.SL.4.1.B Follow agreed-upon rules for discussions and carry out assigned roles.
CCSS.ELA-LITERACY.SL.4.1.C Pose and respond to specific questions to clarify or follow up on information, and make comments that contribute to the discussion and link to the remarks of others.
CCSS.ELA-LITERACY.SL.4.1.D Review the key ideas expressed and explain their own ideas and understanding in light of the discussion.
CCSS.ELA-LITERACY.SL.4.3 Identify the reasons and evidence a speaker provides to support particular points.

Language:

CCSS.ELA-LITERACY.L.4.1 Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.
CCSS.ELA-LITERACY.L.4.1.A Use relative pronouns (*who, whose, whom, which, that*) and relative adverbs (*where, when, why*).
CCSS.ELA-LITERACY.L.4.2 Demonstrate command of the conventions of standard English capitalization, punctuation, and spelling when writing.
CCSS.ELA-LITERACY.L.4.3 Use knowledge of language and its conventions when writing, speaking, reading, or listening.
CCSS.ELA-LITERACY.L.4.3.A Choose words and phrases to convey ideas precisely.
CCSS.ELA-LITERACY.L.4.4 Determine or clarify the meaning of unknown and multiple-meaning words and phrases based on grade 4 reading and content, choosing flexibly from a range of strategies.

Grade 5

Reading Informational Text:

CCSS.ELA-LITERACY.RI.5.2 Determine two or more main ideas of a text and explain how they are supported by key details; summarize the text.
CCSS.ELA-LITERACY.RI.5.3 Explain the relationships or interactions between two or more individuals, events, ideas, or concepts in a historical, scientific, or technical text based on specific information in the text.
CCSS.ELA-LITERACY.RI.5.4 Determine the meaning of general academic and domain-specific words and phrases in a text relevant to a *grade 5 topic or subject area*.

Writing:

CCSS.ELA-LITERACY.W.5.1 Write opinion pieces on topics or texts, supporting a point of view with reasons and information.
CCSS.ELA-LITERACY.W.5.1.A Introduce a topic or text clearly, state an opinion, and create an organizational structure in which ideas are logically grouped to support the writer's purpose.
CCSS.ELA-LITERACY.W.5.1.B Provide logically ordered reasons that are supported by facts and details.
CCSS.ELA-LITERACY.W.5.1.C Link opinion and reasons using words, phrases, and clauses
CCSS.ELA-LITERACY.W.5.1.D Provide a concluding statement or section related to the opinion presented.
CCSS.ELA-LITERACY.W.5.2 Write informative/explanatory texts to examine a topic and convey ideas and information clearly.
CCSS.ELA-LITERACY.W.5.2.A Introduce a topic clearly, provide a general observation and focus, and group related information logically; include formatting (e.g., headings), illustrations, and multimedia when useful to aiding comprehension.
CCSS.ELA-LITERACY.W.5.2.B Develop the topic with facts, definitions, concrete details, quotations, or other information and examples related to the topic.
CCSS.ELA-LITERACY.W.5.2.C Link ideas within and across categories of information using words, phrases, and clauses (e.g., *in contrast*, *especially*).
CCSS.ELA-LITERACY.W.5.2.D Use precise language and domain-specific vocabulary to inform about or explain the topic.
CCSS.ELA-LITERACY.W.5.2.E Provide a concluding statement or section related to the information or explanation presented.
CCSS.ELA-LITERACY.W.5.4 Produce clear and coherent writing in which the development and organization are appropriate to task, purpose, and audience.
CCSS.ELA-LITERACY.W.5.8 Recall relevant information from experiences or gather relevant information from print and digital sources; summarize or paraphrase information in notes and finished work, and provide a list of sources.

Speaking & Listening:

CCSS.ELA-LITERACY.SL.5.1 Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on *grade 5 topics and texts*, building on others' ideas and expressing their own clearly.
CCSS.ELA-LITERACY.SL.5.1.A Come to discussions prepared, having read or studied required material; explicitly draw on that preparation and other information known about the topic to explore ideas under discussion.
CCSS.ELA-LITERACY.SL.5.1.B Follow agreed-upon rules for discussions and carry out assigned roles.
CCSS.ELA-LITERACY.SL.5.1.C Pose and respond to specific questions by making comments that contribute to the discussion and elaborate on the remarks of others.
CCSS.ELA-LITERACY.SL.5.1.D Review the key ideas expressed and draw conclusions in light of information and knowledge gained from the discussions.
CCSS.ELA-LITERACY.SL.5.3 Summarize the points a speaker makes and explain how each claim is supported by reasons and evidence.

Language:

CCSS.ELA-LITERACY.L.5.1 Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.
CCSS.ELA-LITERACY.L.5.1.A Explain the function of conjunctions, prepositions, and interjections in general and their function in particular sentences.
CCSS.ELA-LITERACY.L.5.2 Demonstrate command of the conventions of standard English capitalization, punctuation, and spelling when writing.
CCSS.ELA-LITERACY.L.5.3 Use knowledge of language and its conventions when writing, speaking, reading, or listening.
CCSS.ELA-LITERACY.L.5.4 Determine or clarify the meaning of unknown and multiple-meaning words and phrases based on grade 5 reading and content, choosing flexibly from a range of strategies.

Common Core Math Standards (Grades 3 – 5)

Grade 3

MP.1 Make sense of problems and persevere in solving them.

MP.2 Reason abstractly and quantitatively.

MP.4 Model with mathematics.

MP.6 Attend to precision.

3.OA.1-4 Represent and solve problems involving multiplication and division.

3.OA.5-6 Understand properties of multiplication and the relationship between multiplication and division.

3.OA.7 Multiply and divide within 100.

3.NBT.1 Use place value understanding and properties of operations to perform multi-digit arithmetic.

3.MD.3-4 Represent and interpret data.

Grade 4

MP.1 Make sense of problems and persevere in solving them.

MP.2 Reason abstractly and quantitatively.

MP.4 Model with mathematics.

MP.6 Attend to precision.

4.OA.1-3 Use the four operations with whole numbers to solve problems.

4.OA.5 Gain familiarity with factors and multiples.

4.NBT.4-5 Use place value understanding and properties of operations to perform multi-digit arithmetic.

4.MD.4 Represent and interpret data.

Grade 5

MP.1 Make sense of problems and persevere in solving them.

MP.2 Reason abstractly and quantitatively.

MP.4 Model with mathematics.

MP.6 Attend to precision.

5.MD.2 Represent and interpret data.

5.MD.1-2 Graph points on the coordinate plane to solve real-world and mathematical problems.

Additional Resources

Videos

https://www.youtube.com/watch?v=yHGODp0b8Ks A fun little video about an "invisible" computer mouse. (3:05 min)
https://www.youtube.com/watch?v=lvtfD_rJ2hE An interesting video demonstrating a new method of human computer interface (3:40 min). Short Ad at start.
https://www.youtube.com/watch?v=bQilo5ecSX4 Introduction to the "Hour of Code". (2 min) Includes a short introduction to Blockly which is similar to SCRATCH.
https://www.youtube.com/watch?v=OWJCfOvochA Interesting video on quantum computing. Might be a bit advanced for students but interesting. Good for advanced students. (20 min) Short ad to start.

DK introduction to coding videos:

https://www.youtube.com/watch?v=THOEQ5soVpY 1:55 min
https://www.youtube.com/watch?v=Nc31NAujTkA 2:32 min
https://www.youtube.com/watch?v=xngWoocXYCo 2:48 min

Timeless app videos:

https://www.youtube.com/watch?time_continue=91&v=hR8I66p66mU (a 3:45 min video about the app)

https://www.youtube.com/watch?v=olZTPRbXGw4 (a 4:10 min video about Emma and what motivates her)

https://www.youtube.com/watch?v=wOKJ0-jOSyY (30 min+ interview/talk by Emma at Google)

Alex Deans, iAid videos

www.youtube.com/watch?v=EGPo7gnvlhE&t=1s. Short YouTube video in which Alex demonstrates his device.
https://www.ted.com/talks/alex_deans_life_lessons_of_a_teen_inventor?language=en This is an absolutely wonderful TEDx talk by Alex. It is a bit long (9 min) but very rich in content to build the confidence of young learners that it is worth the time.

Roy Allela videos

https://youtu.be/DeNjwKzTaiQ

Websites

http://www.raspberrypi.org/ Raspberry PI is a more sophisticated version of the BBC micro:bit. A great next step.
https://scratch.mit.edu/ Scratch website

Circuits and Electricity

Videos

https://www.youtube.com/watch?v=VnnpLaKsqGU Video (2:27) on basics of an electrical circuit. You have the option of skipping a video before video starts.

http://www.youtube.com/watch?v=20Vb6hlLQSg Video (5:18 min) on energy generation and distribution.

https://www.youtube.com/watch?v=1-g73ty9v04 (3:50 min) A fun short video that illustrates many ways to save energy. This is a good starting point for a discussion on electricity and energy.

https://www.youtube.com/watch?v=6p5WXzrYYiI A video on the history of electricity as told by an 8th grader (5:25 min)

http://www.neok12.com/video/Electricity/zX7772537442630f465e4751.htm A short video on electricity (5:15 min)

Articles

http://electronics.howstuffworks.com/led.htm Excellent teacher article on how LEDs work.

http://www.tvakids.com/electricity/conservation.htm Article on energy efficiency for kids.

Websites

http://www.neok12.com/Electricity.htm A website with games and videos about electricity.

https://www.mitel.com/en-us/articles/history-telecommunication A good summary about the history of telecommunication (communication over long distances).

Appendix

Contents

Electronic copies of all notebooks and student sheets can be found on the unit digital forum.

Accessing Online Content

To better support you in the implementation of this curriculum we suggest the following:

- Please review our short tutorial video at https://creosityspace.com/resources/

- To access the digital materials, please register your unit by emailing kath@creosityspace.com with registration code provided in the front page of this educator guide. At that point you can also schedule a complimentary online information session to support implementation of the *Coding Whisperers* unit in your classroom.

Science Safety and Behavior Contract

I know the class emergency plan

- If anything happens, the first thing you need to do is tell the teacher.
- If something dangerous happens, we will walk outside into the hall to make sure everyone is safe.

I will follow directions

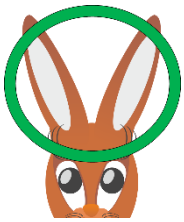
- Make sure that you know what to do before you do it. Ask any questions BEFORE you begin.

I will wear goggles if I am told to

I will tell the teacher if I have an accident or injury

- Even if it seems small, please let the teacher know if an accident or injury happens.

I will listen carefully



I understand I will be removed from the science activity area by the teacher if **I am preventing others from learning.**

I will not touch any materials unless I have been given permission

I will wash my hands after science activities

I will not eat, taste, drink, or inhale anything we use in science

I have read the attached safety rules and have been present when they were discussed in class.

Student's Name _____ Date _____

I have read and discussed the laboratory safety rules with my child.

Parent signature _____ Date _____

This student has allergies/sensitivities to:

_____

Technology Historical Timeline

The Technology Historical Timeline is a great tool to use to give your students historical context for what they are learning and to show them how any single scientific discovery or understanding is built from all the discoveries and understandings that came before. In many cases, scientific discoveries only thrived if there was a community need they helped to solve.

The following worksheets contain a couple versions of the timeline you can use with your class:

- The completed timeline is one we think links critical events in the history of numbers and computing. As an activity you could have your students pick their top three events on the timeline and then justify why they thought they were the most important.

- You could have your students pick one event on the timeline (e.g., the definition of prime numbers) and create another timeline that includes that invention. Some examples could be:

    - How have prime numbers been used throughout the ages
    - Major advances in math over the years
    - Things that are grouped together because of a trait they don't have (e.g. for prime numbers they can't be divided evenly by anything except 1 and themselves)

- You could take the blank timeline and have the students pick a different technology from which to create their own timeline.

- You could take the dateless descriptions and have the students try to put them in chronological order. Ask them to justify their order.

- Most online technology historical timelines have a paragraph associated with each event. As a class you could pick a different technology historical timeline (these can be found online by googling "X historical timeline") and have the students use the blank timeline to summarize and write down critical events.

20,000 BCE

Artifacts discovered in Africa suggests that early people have attempted to measure time.

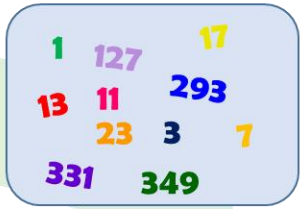
3000 - 2000 BCE

Evidence of the Babylonians (modern day Iraq) using numbers and performing multiplication and division.

Yale Babylon Collection

300 BCE

Euclid defines a Prime number as a number that can only be divided by 1 and itself (e.g. 3, 7, 13, 17).



~ 1940 CE

The first electrical computer, ENIAC, is developed for military applications. It is the size of a full room.

*

100 BCE

The abacus (the first type of calculator) is re-introduced into Europe.

1970 CE

Mathematicians realize that prime numbers can be used to generate decryption (decoding) keys.

1974 CE

Scelbi 8H Combato, the first "personal computer" is developed.

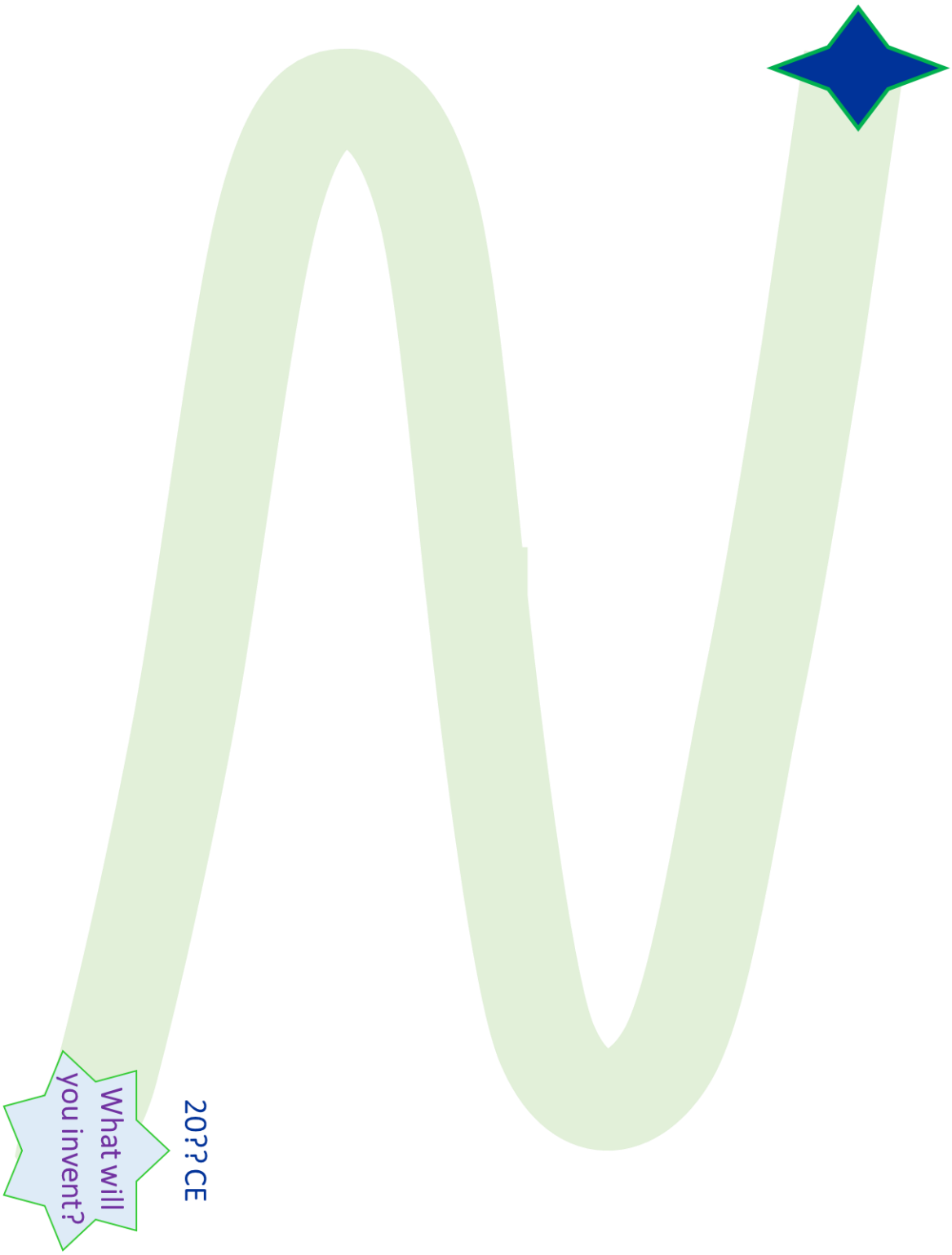
2007 CE

Apple launches the iPhone - which is basically a full computer in your hand

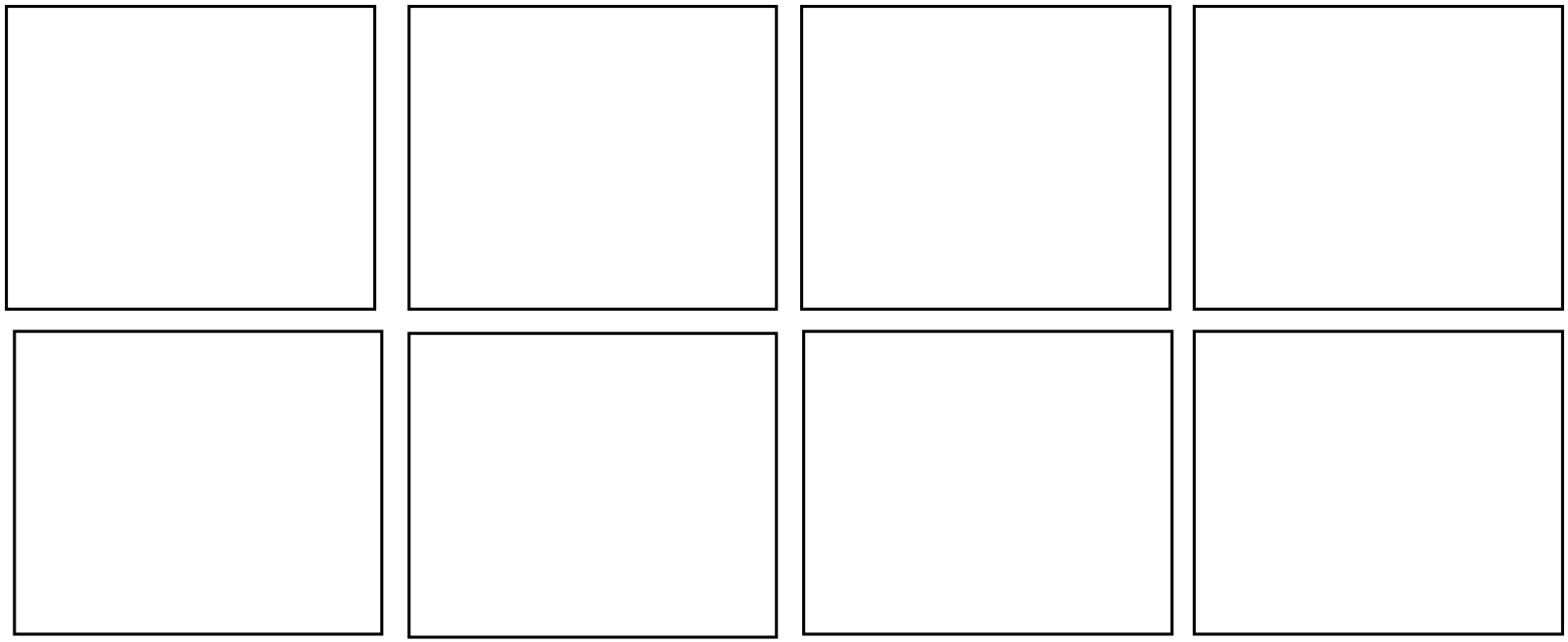
20?? CE

What will you invent?

Copyright University of Pennsylvania



Euclid defines a Prime number as a number that can only be divided by 1 and itself (e.g. 3, 7, 13, 17). 	Mathematicians realize that prime numbers can be used to generate decryption (decoding) keys. 	Evidence of the Babylonians (modern day Iraq) using numbers and performing multiplication and division. 
Scelbi 8H Combato, the first "personal computer" is developed. 		Apple launches the iPhone - which is basically a full computer in your hand 
Artifacts discovered in Africa suggests that early people have attempted to measure time. 	The abacus (the first type of calculator) is re-introduced into Europe. 	The first electrical computer, ENIAC, is developed for military applications. It is the size of a full room. 



Put the historical events in the order you think they belong and justify your order.

Exit Ticket Support

Exit tickets are a great method for quick formative assessments. In some cases, they can also be used a quick pre-assessment or a conversation starter.

As a formative assessment strategy, students should hand you their completed exit tickets as they are transitioning to another activity (e.g., lunch, recess, gym, etc.). Exit tickets can be completed on index cards, half-sheets of paper, etc. and should only take about five minutes for students to complete. Exit tickets help you assess if students have understood the main concepts from the preceding lessons. If not, you may choose to have a additional discussion or to repeat portions of the lesson so that students have additional time to explore and practice key concepts.

What does programming allow you to do?

Programming allows you to get machines or computers to do what you want them to do.

Are computers smart? Why or why not?

In general computers are not smart as they can only follow the specific rules laid out for them. The exception to this is artificial intelligence—although students are not expected to consider artificial intelligence at this time.
(Note: Students who are more familiar with artificial intelligence might struggle with this question because their general knowledge will not align exactly with what is being discussed in class. It is OK if they give a different opinion that includes their extra knowledge.)

Why is it helpful to plan out your program before you start coding?

It is helpful to plan out your program before you start because it is important that a program contains all the necessary details—in the correct order. Planning things out ahead of time can help ensure that you don't forget anything.

Describe physical computing in your own words.

Physical computing is the ability to use computer code to control devices with which we can touch and interact.

What is an LED?

An LED is a light emitting diode. It is a type of light that doesn't generate a lot of extra heat and can but made quite small.

Name three things you could program and how you know they are programmable.

This list can be quite long and is likely to include things like dishwashers, washing machines, coffee makers, DVR, phone, etc. You know something is programmable because it can perform a series of predetermined functions vs just on and off.

My STEM Stories™ Notebook

The next few pages contain copies of the *My STEM Stories*™ notebook.

Please refer to the online content webpage for:

- Versions of this notebook that can be printed single sided. *Note: The single sided version is formatted so that each page is presented as an 8.5 x 11 spread—similar to the layout in the following pages.*
- Version of this notebook that can be printed double sided.
- Digital files with text descriptions of the pictures.
- Digital files that can be easily incorporated into online learning platforms such as Google Classrooms or SeeSaw.

Prior Knowledge Expectations

Students are not expected to come to this unit with any specific understanding around coding. It is expected that all students will have interacted with programmable devices (phones, computer, dishwashers, etc.) even if they may not have recognized that these devices were programmable.

Is there something cool about science, math, engineering, or technology that you want to share with your class and family? Write about it here.

Read about STEM topics other kids find cool at the Kids' Corner on our website.

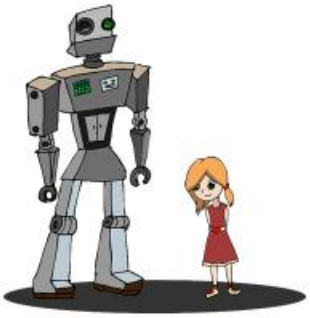
CreositySpace
Connecting Kids to STEM Through Entrepreneurship

My STEM Stories™

Coding Whisperers

Featuring Embedit Electronics

How do you talk to a robot?



Name:

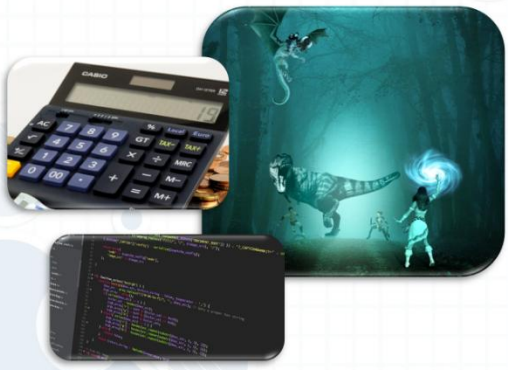
Vocabulary

Term	Definition	A Picture or Simplified Definition
Computer Programming (coding)	Computer programming is a way for a person to translate their logic into a computer language.	
Computer Hardware	Computer hardware are the parts of a computer you can touch—such as the screen, computer chips, and keyboard.	
Computer Software	Computer software, or computer code, is the set of instructions that tell the computer what to do.	
Physical Computing	Physical computing is the ability to control computer hardware and physical devices (motors, sensors, lights) with computer software.	
Algorithm	An algorithm is a guide, or an outline, that is used to lay out all the steps needed for a computer program or code.	
Input/Output (I/O)	Inputs and Outputs allow for the user to control software. Touchscreens, keyboards, mice, and joysticks are all examples of I/O devices.	

2

Coding

Coding, also known as computer programming, is the language of electronics. It can be as straightforward as asking a device to calculate two plus two or as complicated as creating a video game world where the number of outcomes between two players is almost impossible to quantify.



Coding is such a big part of our everyday life that soon babies might learn a computer language at the same time they are learning how to speak!

3

There are many examples of coding all around us. One example is physical computing, which is when computers and electronics interact with the outside world through a series of rules or *preprogrammed commands*. One example you might be familiar with is a video game controller.

With physical computing, the electronics respond to specific instructions that are written in the code. These instructions may come from the words in the code itself or how the code tells the electronics to respond to an external action (like moving a joystick or removing a light source).

Another field of coding is that of data analysis—which is paving the way for the development of new technologies such as artificial intelligence. In these applications computers analyze a variety of inputs and then make their own decisions. These decisions can be a simple as determining the pattern in a series of numbers or as complicated as a self-driving car or a chess-playing robot.

Every day entrepreneurs, engineers, and regular people (like you and me) come up with new ways to use coding to solve problems in their lives. Is there a challenge in your life that you think coding could help you solve?

How is coding like a spoken language?

Describe three ways you interact with coding or computer programming every day?

Create Your Own Timeline

Meet Some Innovators & Entrepreneurs

Timeless (www.timeless.care)

Emma Yang started coding when she was only six and began developing the Timeless app when she was only 8-years old.

At first, she focused her coding efforts on making simple video games but one day she noticed things didn't seem quite right with her grandmother, who lived in Hong Kong. It started when her grandmother forgot her birthday. Then she started calling the house over and over again not realizing they had spoken on the phone a few minutes earlier. It was around this time that Emma's family discovered her grandmother had Alzheimer's disease—a disease that effects your brain and makes it hard to remember names, faces, recent activities, or upcoming plans. Emma wanted to help her grandmother, so she started to do some research.

She figured out what the hardest tasks were for people with Alzheimer's and learned what they needed most. In addition to help with remembering things, one of the most important needs for Alzheimer's patients is that they stay connected with their family and community. In 2019, she launched the Timeless app on the App Store. Timeless focused on keeping people connected while addressing some of the most common challenges. Some other features of the app included daily reminders, photo grouping, and recognition (using artificial intelligence), as well as alerts for things that might be forgotten such as repeated phone calls.

Recently, as Emma started college, she realized she needed to step back from working on and promoting Timeless to focus on school. As a result, there was no one to continue updating the Timeless code and she decided to remove it from the App Store. Despite this transition for her business, Emma remains passionate about using technology to support our aging community and the Timeless website remains an active record of her contribution to this field.

Innovator Showcase

Roy Allela and Sign-IO

Roy Allela is from Kenya. He studied engineering and computer science in college. His 6-year-old niece is deaf, and he wanted to help her communicate. She knows sign language, but many people cannot understand sign language.

Roy developed a glove that his niece could wear that had **sensors** in each finger. These sensors can tell how much the finger was bent and how it was moving. When the gloves are paired with a cell phone or tablet, much the same way we pair wireless headphones, and the phone or tablet turns the motion into words. Then, the phone or table can "speak" using a text-to-speech program.

Although Sign-IO gloves are not perfect, they are a great tool for people, especially younger people, who need help with everyday communication.

Entrepreneur Flashcard

Brian Saab, Amy Hutchinson & Nate Miller at UNEARTH Technologies

Meet Nate, Amy, and Brian. They started a company called Unearth Technologies. They each had different skills and interests. Brian liked business and management experience. Amy had software and product development experience. Nate was focused on visual design and understanding customers.

Computers, cell phones, satellites and drones can collect a lot of information—so much information that it can be hard to find what you need (like finding matching socks in a huge pile of laundry!).

That's where Unearth Technologies made a difference. They were experts in both COLLECTING and ORGANIZING information. Unearth Technologies designed special software to help the construction industry to help them keep track how far along projects were and where everyone and everything was.

In 2023 Unearth Technologies was acquired by Procore and Nate, Amy and Brian moved on to their next adventure. Congratulations to all of them!

Innovator Spotlight

Alexander Deans and the iAid Device

Alexander Deans, a 16-year-old from Ontario, Canada, developed the iAid device to help people who are blind or have trouble seeing get around more easily. He designed a joystick device that can be hooked up to your phone and helps the user move around without hitting things. Listen to Alex describe his innovation in this video www.youtube.com/watch?v=EGPo7gnvlhE&t=1s.

How are innovators and entrepreneurs using coding to help people? Describe a problem you see in your home, school, or neighborhood, and discuss how coding might be used to help with a solution.

My STEM Explorer Notes™ Notebook

The next few pages contain copies of the *My STEM Explorer Notes*™ notebook.

Please refer to the online content webpage for:

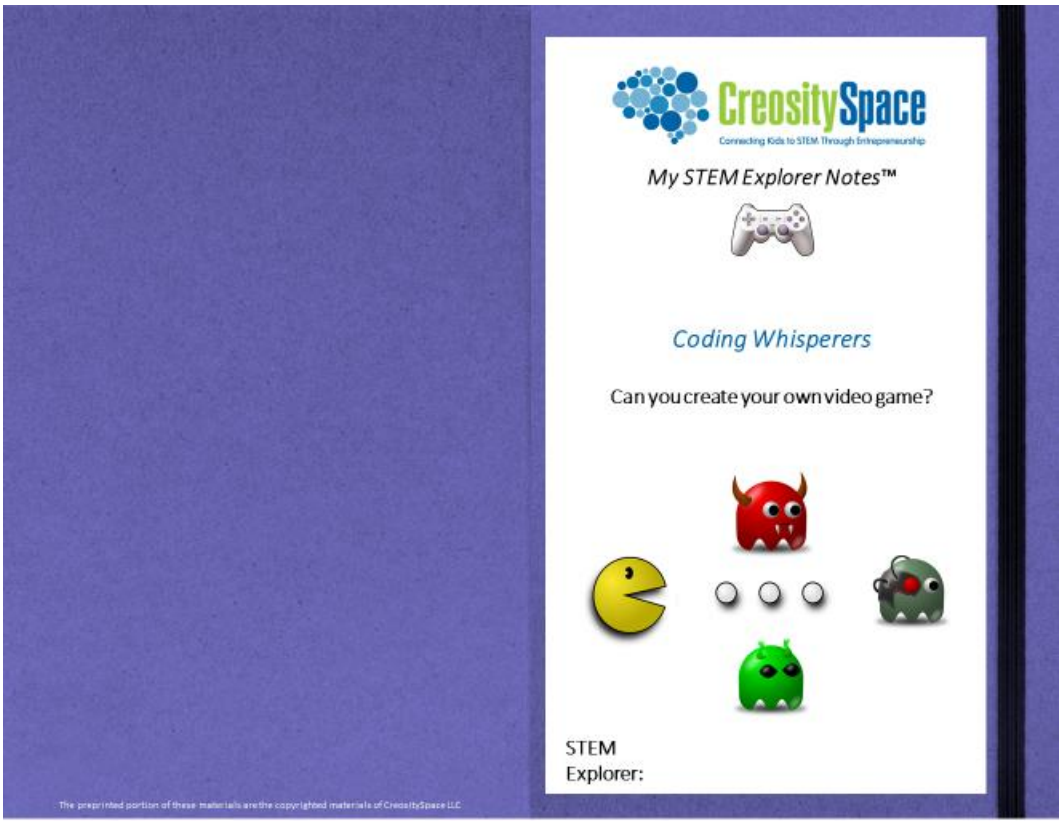
- Versions of this notebook that can be printed single sided. *Note: The single sided version is formatted so that each page is presented as an 8.5 x 11 spread—similar to the layout in the following pages.*
- Version of this notebook that can be printed double sided.
- Digital files with text descriptions of the pictures.
- Digital files that can be easily incorporated into online learning platforms such as Google Classrooms or SeeSaw.

Prior Knowledge Expectations

Students are not expected to come to this unit with any specific understanding around coding.

To use the BBC micro:bit pins it is helpful if students have a general understanding of circuits and electricity. If you need to provide some additional information for your students on circuits, some introductory references are provided in the Additional Resources.

It is also expected that students have some experiences with the engineering design processes, asking questions, developing solutions, and having productive (and respectful) collaborative discussions. To support this there are a number of scaffolds provided in the Teacher Support Documents section of the Appendix and the first few pages of the My STEM Explorer Notes™ notebooks have a series of *Useful Phrases for Having Constructive Discussions*.



Useful Phrases for Having Constructive Discussions

Asking Clarifying Questions

Can you be more specific?
Can you explain your answer further?
Can you give an example?
Can you please explain your thinking?
Can you repeat what you said?
Could you rephrase that?
Could you say that one more time?
What is your evidence?
Can you give me another example, so I can understand?
Can you tell me more?
Why do you think that is important?
Why do you think that happened?
What if the opposite were true?

2

Adding to an Idea

I agree with _____ because _____.
I agree with _____.
I agree with _____ and I also think _____.
I agree with _____ and, would like to add _____.
I agree, and I have an addition: _____.
I believe this is true because _____.
I know that too because _____.
I have something to add; _____.
I think you are right, and I also think _____.
I would like to add to that idea.
This reminds me of _____ because _____.
Yes, that makes sense, and I would also like to add _____.

Respectfully Disagreeing with an Idea

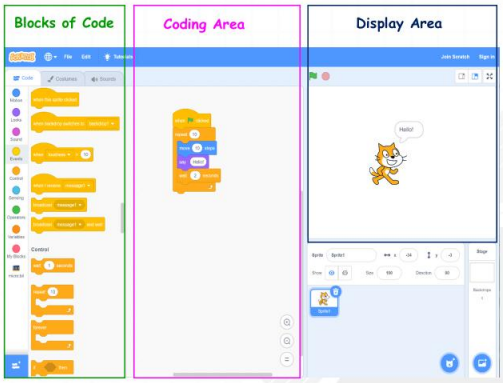
Could you explain, because I have a different idea.
I disagree with that idea because _____.
I disagree with your reasoning because _____.
I disagree with _____ because _____.
I have completely different opinion on that.
I respect your opinion and _____.
I respect your point, and in my opinion _____.
I respectfully disagree because _____.
I see your reasoning and disagree with some of the idea because _____.
That's a good point, and _____.

3

Welcome to SCRATCH!

SCRATCH is a *drag-and-drop* coding program developed by people at MIT. SCRATCH allows everyone to write code using *blocks*, instead of words and symbols, so you do not have to worry about getting the coding *grammar* (called syntax) perfect.

This is your SCRATCH workspace.



To write a program you drag **blocks of code** into the **coding area** and connect them. When you click on your program in the **coding area** it plays in the **display area**.

There are a lot of different things you can do with SCRATCH. Let's warm-up by having Scratch the cat perform a few tasks.

Task 1

1. Move Scratch the cat forward 10 steps
2. Turn Scratch the cat 15 degrees
3. Have Scratch the cat play the sound meow

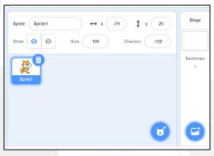
(Hint: To get the code to start, click directly on it.)

Task 2

Have Scratch the cat complete the above sequence ten times with a one second wait between each repeat.

Task :

Can you move Scratch the cat back to the top left corner and have him stand up straight? *(Hint: Use the mouse and/or the controls in the bottom right corner.)*

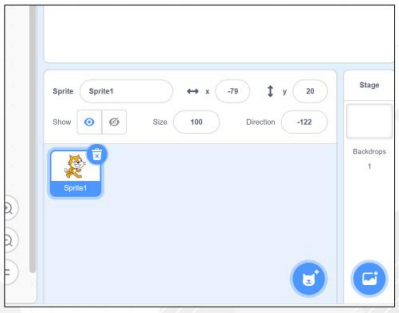


Task 4

Can you put an outfit onto Scratch the cat or turn him into another animal? *(Hint: Look under the Costume tab.)*

Task 5

Can you add another character or sprite? Can you change the background? *(Hint: Look at the options in the bottom right corner.)*



Use this space to write down some notes about using SCRATCH.

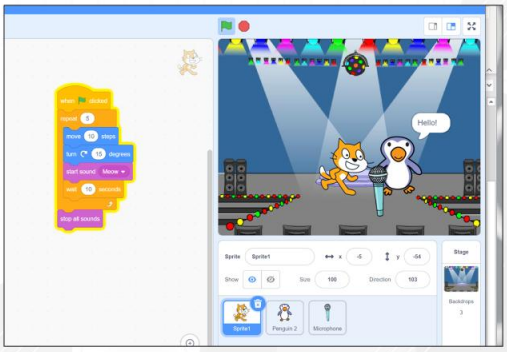
1 Click on the green flag or the code to start the program.
2 Click on the stop sign to stop the program.
3 To disconnect blocks, you must pull them down.
4 Each sprite has their own code window.
5 Drag code to the column on the left to get rid of it.
6
7
8
9
10
11
12
13
14
15
16

Animation Time

Now it is time to create your own animation. You get to choose the details of your animation, but it must include the following things:

- ✓ At least two characters (or sprites)
- ✓ Some motion
- ✓ Some sound
- ✓ A background that is something other than white

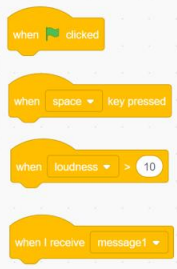
Here is an example.



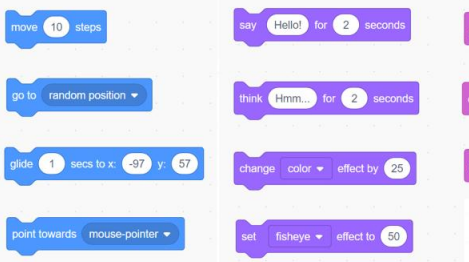
You can use this space to plan out your animation or you can dive right in. The next page has some useful blocks of code you can use.

Some Useful SCRATCH Blocks

Getting Started



One Step Actions



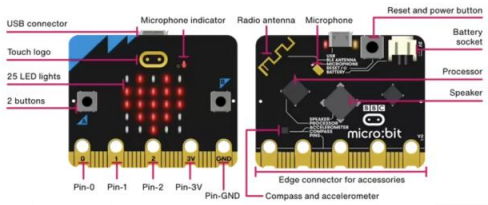
More Advanced Actions

These are just some of the SCRATCH blocks you can use. There are many more waiting for you to test them out!

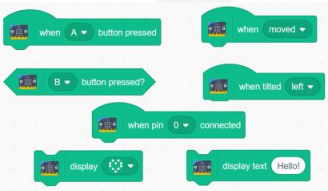
Introducing the BBC micro:bit

The BBC micro:bit is like a fancy controller that you can program to do a lot of different things. It has buttons (like a video game controller), a motion sensor (like airplane controls), and can act like a communication bridge between your SCRATCH code and the outside world.

The layout of the BBC micro:bit



You can control the micro:bit in SCRATCH by dragging different blocks into the coding area. Here are some of the mirco:bit blocks.

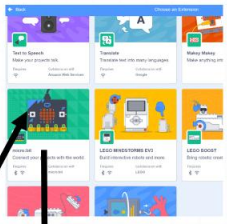


Making sure SCRATCH can see the BBC micro:bit

It is important to make sure SCRATCH and the micro:bit are properly connected.

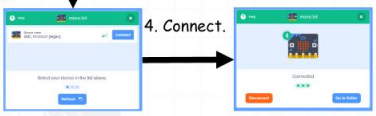
1. Select the extensions tab.

2. Select the micro:bit extension.



3. Identify your micro:bit.

4. Connect.



If you run into problems, there are some troubleshooting suggestions on the last two pages.

Let's get some practice using the BBC micro:bit.

The LED display

Can you...

- light up a heart on the LED display?
- have the LED display show your name?
- change the LED display by connecting a pin?

Controlling a sprite (or character)

Can you...

- control Scratch the cat by pressing a button on the micro:bit?
- control Scratch the cat by moving the micro:bit?
- control Scratch the cat with a command started by a pin?

Integrating the micro:bit into your animation

Can you...

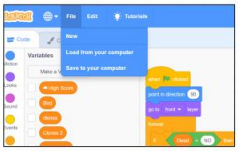
- use either the A or B button in your animation?
- use the *jump* or *shake* function in your animation?
- use pin 2 in your animation?

Use this space to write down some notes using your BBC micro:bit.

1 Never connect the ground and 3V pins.
2
3
4
5
6
7
8
9
10
11
12
13

Can You Hack Flappy Bird?

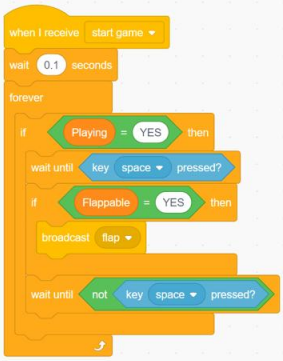
To load Flappy Bird, go to ***File/Load from your computer*** and select the **Flappy Bird Scratch Edition.sb3**. It should be on your desktop.

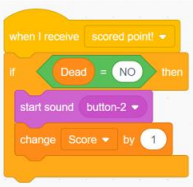


By changing the code, you can change the way the game is played (and scored). Can you make the following changes?

1. Ensure that you get a new high score as soon as you clear the first obstacle.
2. Make it easier for the Flappy Bird sprite to get through the obstacles.
3. Change how the Flappy Bird sprite flaps:
   - Use a different keyboard key
   - Use a motion on the BBC micro:bit
   - Use a button on the BBC micro:bit
4. Can you add something to the Flappy Bird code that shows your score on the BBC micro:bit display?

Here are some pieces of the Flappy Bird code that might help you with your modifications.





You can use this space to keep track of your favorite hacks.

Time to Design Your Video Game

Before you jump into designing your video game it is helpful to do some planning for the game's storyline. You can revise your plan as you go along but it is important to have a place to start.

Some questions you should think about are:

- ✓ What is the goal? How do you win?
- ✓ Where does the game take place?
- ✓ How many characters are in the game?
- ✓ How does the game end—do you need to win, or can you die?

What are some other details about the storyline of your video game?

You can use the space on the next few pages to sketch out your video game. You might want to go back and forth between your sketches and your code—or not—either way is okay, just pick a strategy that works for you!

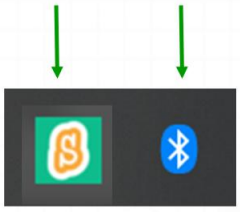
24

25

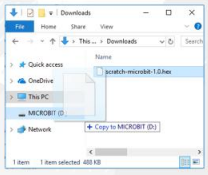
Troubleshooting the BBC micro:bit with SCRATCH

Make sure the BBC micro:bit has power through a USB connection to the computer.

Make sure SCRATCH link and Bluetooth are active.

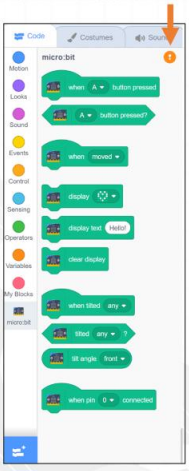


Make sure the SCRATCH micro:bit hex is installed on your BBC micro:bit.

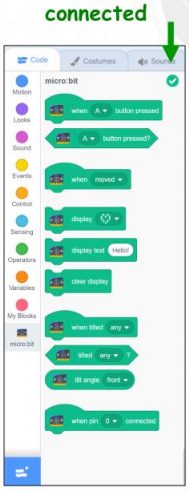


Make sure SCRATCH sees the BBC micro:bit.

Orange means NOT connected



Green means connected



Go to www.scratch.mit.edu/microbit for download links and more details.

26

27

Teacher Support Documents

The following resources have been recommended to us by a number of teachers who are a part of our CreositySpace community. While many of them are freely available via a quick internet search, we have included them here for your use. Where possible, we have given credit to the resource's creator.

1. Useful phrases for having constructive discussions
2. Goals for Productive Discussions and Nine Talk Moves (excerpt from TERC's Talk Science Primer, S. Michaels and C. O'Connor)
3. Cooperative learning strategies (A. Venegas)
4. ELPS scaffolds (N. Balayan, 2019)
5. Multidimensional strategies that support English language development
6. Claim-Evidence-Reasoning (C-E-R) student graphic organizer (A. Venegas)
7. Supporting students so that they will be more successful at constructing evidence-based explanations (A. Venegas)
8. Modeling support
9. Graph paper
10. Cross-curricular integration w/template (Teaching From the Inside Out) (T. Paradis)
11. Tips for getting student-directed conversations back on topic

Useful Phrases for Having Constructive Discussions

Asking Clarifying Questions

Can you be more specific?

Can you explain your answer further?

Can you give an example?

Can you please explain your thinking?

Can you repeat what you said?

Could you rephrase that?

Could you say that one more time?

Did I hear you correctly what you said … ?

Did I hear you say … ?

Did I understand you when you said … ?

Is this what you said: _____?

What do you mean by _____?

What's another way you might … ?

What is your evidence?

What resources were used for this project?

Adding to an Idea

I agree with _____because _____.

I agree with _____.

I agree with _____, and I also think _____.

I agree with _____, and would like to add _____.

I agree, but I have an addition: _____.

I believe this is true because _____.

I know that too because _____.

I have something to add; _____.

I think you are right, and I also think _____.

I would like to add to that idea.

This reminds me of _____ because _____.

Yes, that makes sense, and I would also like to add _____.

Respectfully Disagreeing with an Idea

Could you explain, because I have a different idea.

I disagree with that idea because _____.

I disagree with your reasoning because _____.

I disagree with _____ because _____.

I have completely different opinion on that.

I respect your opinion and _____.

I respect your point, and in my opinion _____.

I respectfully disagree because _____.

I see your reasoning and disagree with some of the idea because _____.

That's a good point, and _____.

Goals for Productive Discussions and Nine Talk Moves

Goal: Individual students share, expand and clarify their own thinking

1. Time to Think:
   - Partner Talk
   - Writing as Think Time
   - Wait Time
2. Say More:
   - "Can you say more about that?" "What do you mean by that?" "Can you give an example?"
3. So, Are You Saying…?:
   - "So, let me see if I've got what you're saying. Are you saying…?" (always leaving space for the original student to agree or disagree and say more)

Goal: Students listen carefully to one another

4. Who Can Rephrase or Repeat?
   - "Who can repeat what Javon just said or put it into their own words?" (After a partner talk) "What did your partner say?"

Goal: Students deepen their reasoning

5. Asking for Evidence or Reasoning:
   - "Why do you think that?" "What's your evidence?" "How did you arrive at that conclusion?" "Is there anything in the text that made you think that?"
6. Challenge or Counterexample:
   - "Does it always work that way?" "How does that idea square with Sonia's example?" "What if it had been a copper cube instead?"

Goal: Students think with others

7. Agree/Disagree and Why?:
   - "Do you agree/disagree? (And why?)" "Are you saying the same thing as Jelya or something different, and if it's different, how is it different?" "What do people think about what Vannia said?" "Does anyone want to respond to that idea?"
8. Add On:
   - "Who can add onto the idea that Jamal is building?"
   - "Can anyone take that suggestion and push it a little further?"
9. Explaining What Someone Else Means:
   - "Who can explain what Aisha means when she says that?" "Who thinks they could explain in their words why Simon came up with that answer?" "Why do you think he said that?"

Cooperative Learning Strategies

There are some popular strategies that can be used with all students to learn content (such as science, math, social studies, language arts, and foreign languages). However, they are particularly beneficial to ELLs for learning English and content at the same time. Most of these strategies are especially effective in teams of four.

1. Round Robin
   Present a category (such as names of mammals) for discussion. Have students take turns going around the group and naming items that fit the category.

2. Roundtable
   Present a category (such as words that begin with b). Have students take turns writing one word at a time.

3. Write-Around
   For creative writing or summarization, give a sentence starter (for example, if you give an elephant a cookie, he's going to ask for...). Ask all students in each team to finish that sentence. Then, they pass their paper to the right, read the one they received, and add a sentence to that one. After a few rounds, four great stories or summaries emerge. Give children time to add a conclusion and/or edit their favorite one to share with the class.

4. Numbered Heads Together
   Ask students to number off in their teams from one to four. Announce a question and a time limit. Students put their heads together to come up with an answer. Call a number and ask all students with that number to stand and answer the question. Recognize correct responses and elaborate through rich discussions.

5. Team Jigsaw
   Assign each student in a team one fourth of a page to read from any text (for example, a social studies text), or one fourth of a topic to investigate or memorize. Each student completes his or her assignment and then teaches the others or helps to put together a team product by contributing a piece of the puzzle.

6. Tea Party
   Students form two concentric circles or two lines facing each other. You ask a question and students discuss the answer with the student facing them. After one minute, the outside circle or one line moves to the right so that students have new partners. Then pose a second question for them to discuss. Continue with five or more questions.

After each cooperative learning activity, you will want to debrief with the children by asking questions such as: What did you learn from this activity? How did you feel working with your teammates? If we do this again, how will you improve working together?

ELPS Scaffolds

Level 1	Level 2	Level 3	Level 4	Level 5
WHOLE CLASS:	WHOLE CLASS:	WHOLE CLASS:	WHOLE CLASS:	WHOLE CLASS:
Minilesson: - Visuals (video) - Stop and ask questions - Wait time - Private reasoning - Clarify vocab - Graphic organizer - Collecting feedback - TPR/realia - Modeling (verbally, in writing, ELMO) - Multiple representations - Analyzing sample work	Minilesson: - Visuals (video) - Stop and ask questions - Wait time - Private reasoning - Clarify vocab - Graphic organizer - Collecting feedback - TPR/realia - Modeling (verbally, in writing, ELMO) - Multiple representations - Analyzing sample work	Minilesson: - Visuals (video) - Stop and ask questions - Wait time - Private reasoning - Clarify vocab - Graphic organizer - Collecting feedback - TPR/realia - Modeling (verbally, in writing, ELMO) - Multiple representations - Analyzing sample work	Minilesson: - Visuals (video) - Stop and ask questions - Wait time - Private reasoning - Clarify vocab - Graphic organizer - Collecting feedback - TPR/realia - Modeling (verbally, in writing, ELMO) - Multiple representations - Analyzing sample work	Minilesson: - Visuals (video) - Stop and ask questions - Wait time - Private reasoning - Clarify vocab - Graphic organizer - Collecting feedback - TPR/realia - Modeling (verbally, in writing, ELMO) - Multiple representations - Analyzing sample work
DIFFERENTIATION:	DIFFERENTIATION:	DIFFERENTIATION:	DIFFERENTIATION:	DIFFERENTIATION:
Group/pair work: - Intentional grouping - Graphic organizer - Structured talk - Sentence frames - Translations - Pictures - Word bank - Multiple pathways Individual work: - Translations - Sentence frames - Graphic organizer - Vocabulary - Different ways to show what they know - Extended time - Making connections between representations	Group/pair work: - Intentional grouping - Graphic organizer - Structured talk - Sentence frames - Translations - Pictures - Multiple pathways Individual work: - Translations - Sentence frames - Vocabulary - Graphic organizer - Different ways to show what they know - Making connections between representations	Group/pair work: - Intentional grouping - Graphic organizer - Structured talk - Sentence frames - Multiple pathways Individual work: - Graphic organizer - Vocabulary - Making connections between representations	Group/pair work: - Intentional grouping - Graphic organizer - Structured talk - Multiple pathways Individual work: - Graphic organizer - Making connections between representations	Group/pair work: - Intentional grouping - Graphic organizer - Structured talk - Multiple pathways Individual work: - Graphic organizer - Making connections between representations

Multidimensional Strategies That Support English Language Development

When planning for instruction, use a variety of strategies, techniques, and materials for making grade-level core curriculum accessible for English language learners while at the same time promoting their English language development. The chart below provides examples of sensory, graphic, and interactive supports for English language development within each lesson. Use at least one strategy from each column daily (for example, when showing videos, use the graphic organizer to take notes in addition to providing students with an opportunity to turn and talk with partners).

Sensory Support	Graphic Support	Interactive Support
Real-life objects	Charts	In pairs or partners
Scientific instruments	Graphs	Small groups
Measurement tools	Tables	Whole group
Physical models	Number lines	Using cooperative group structures
Natural materials	Timelines	Using the internet or software programs
Actual substances	Advanced organizers	In student's native language
Organisms or object of investigation	Drawing	With mentors
Posters/illustrations of processes or cycles	Models	Other
Illustrations and diagrams	Graphic organizers (Venn diagram, T-chart, cycles, cause and effect, semantic web)	
Pictures, icons, and symbols	Other	
Videos and films		
Interactive investigations		
Photographs		

Support is an instructional strategy or tool used to assist students in accessing content necessary for classroom understanding or communication. Support may include teaching techniques, such as modeling, feedback, or questioning. Other types of support involve students using visuals or graphics, interacting with others, or using their senses to help construct meaning of oral or written language. We believe that support is important for all learners to gain access to meaning through multiple modalities, but it is absolutely essential for ELL.

Claim-Evidence-Reasoning (C-E-R) Student Graphic Organizer

Question: _____

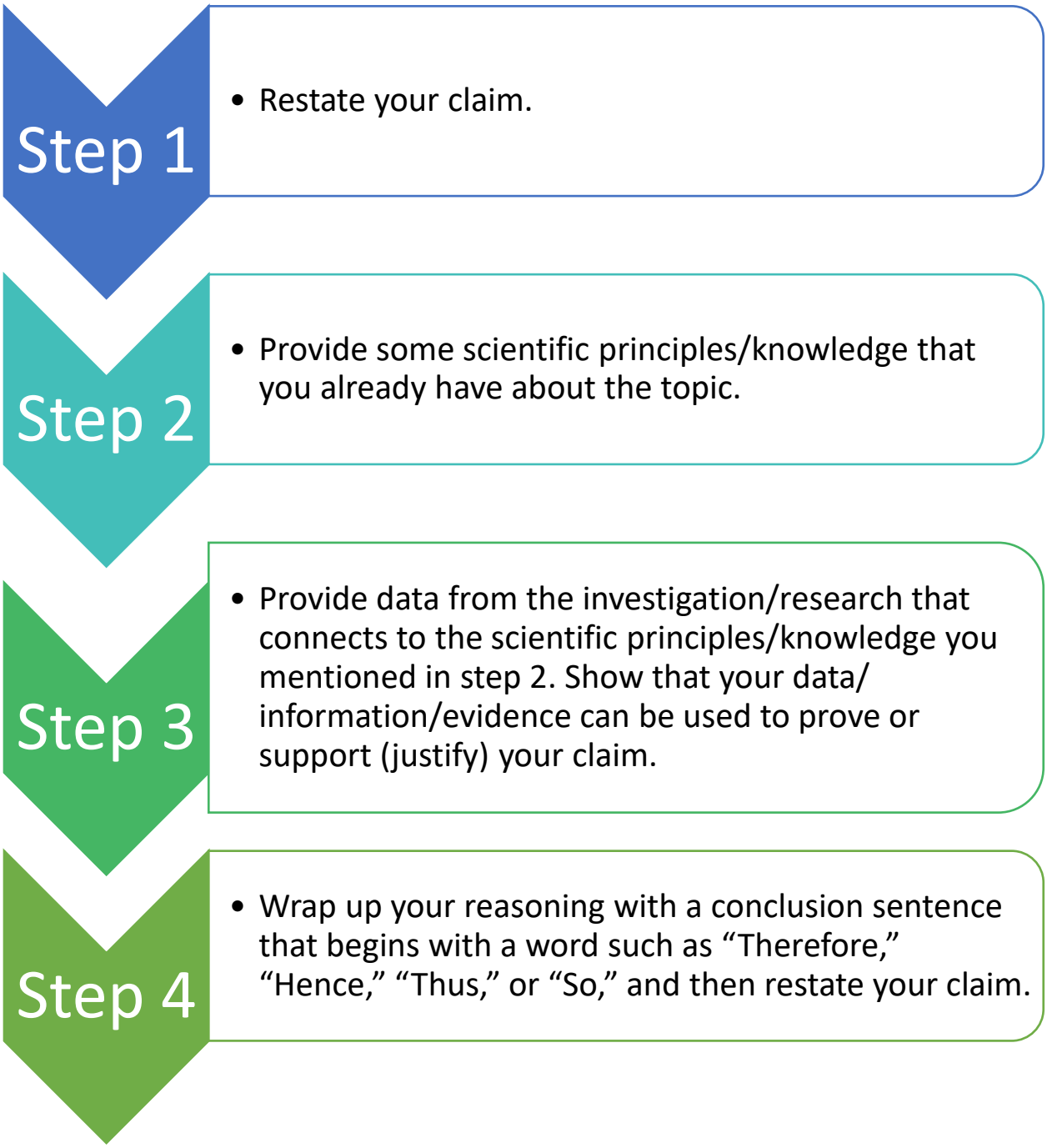
C (Claim) Write a statement that responds to the question.	

E (Evidence) Provide scientific data to support your claim. Your evidence should be appropriate (relevant) and sufficient (enough to convince someone that your claim is correct). This can be bullet points instead of sentences.	

R (Reasoning) Use scientific principles and knowledge that you have about the topic to explain **why** your evidence (data) supports your claim. In other words, explain how your data proves your point. (Use paragraph format.)	

Need help with your reasoning?

Follow this path….



Supporting Students So That They Will Be More Successful at Constructing Evidence-Based Explanations

Setting students up for success

To help students successfully construct evidence-based explanations, it is imperative that you provide support in time for formulating ideas before you ask students to formally talk or write about their final explanations in front of "high-stakes" audiences such as you and the whole class. The following steps have helped me ensure that students have the ideas, skills, language, and confidence necessary for success in my science classes. Everything I've included here came from 10 years of collaboration with numerous specialist at schools where I've worked. When you have a student with special needs or ELL students, I recommend talking about specific strategies that you can try in your classroom. No strategy will work like magic, but overtime you will find ways to support learning for every student.

Step 1: Structured rehearsal time to prethink and prewrite
[This usually last for two to three days about a week before I expect students to have a polished version of their explanations to share publicly.]

STRUCTURED PRETHINKING AND PREWRITING

Students need lots of structure, scaffolding, and time to think about their ideas and to write or draw their ideas in small pieces.

- Prompt students to write or draw their explanation in narrative form like a storyboard. Students should be telling or showing their ideas about the beginning, middle, and end of the phenomenon. For example:

Beginning:	Middle:	End:

WRITING SCAFFOLDS

Use writing scaffolds to support students' writing in the bottom row of their storyboard. I provide a mixture of generic writing scaffolds that students can use in any explanation for any science idea and specific writing scaffolds are only useful in the context of one particular assignment . For example:

1) Sentence starters—help students by giving them a running start for some sentences.
    a. Examples: "The rollerblader starts his gliding motion by…." "One reason this happens is because…." "This made me think that…."

2) Sentence idea banks —help students by prompting specific science ideas for a particular explanation.
    a. Examples: “In your explanation, be sure to use these ideas: push, drag, normal force, friction force,….”
3) Sentence transition phrases—help students by providing connecting words/phrases characteristic of academic writing.
    a. Examples: “In addition,….” “Therefore,….” “But another possibility is….” “Because….”
4) Science concept cards—help students utilize science terminology while writing.
    a. Examples: Students build a collection of cards with terms, drawings, and student-generated definitions. Students are free to use these cards whenever they are working on tasks in class or on homework assignments. A Word Wall can serve a similar function.

STRATEGIC PARTNERSHIPS

This prethinking or prewriting task can be done individually or in pairs where the partners have been strategically selected to support the specific needs of certain students. The goal when planning strategic thinking or writing partnerships is to distribute the cognitive load across two people without totally removing the cognitive load from either student. It takes time to build a culture where two students have learned how to be good partners. As students develope throughout the year, this scaffolding can be reduced and eventually removed.

- Consider pairing two same language ELL students where one student has developed slightly more advanced English language skills so that students could switch back and forth between languages. Consider allowing ELL students to communicate in a language other than English and work on translating into English later. However, I would not want students to be excluded from using English or excluded from hearing academic English (students need to use and hear academic English in order to learn academic English). Use these pairings sparingly—introduce this practice as students progress throughout the year.
- Consider pairing certain special needs students with patient and helpful students who can serve as sounding boards, writing tutors, or reminders to stay focused. However, remember that there is a lot to consider here:
    - Students won’t learn to develop literacy skills (like writing) if they are never allowed to write so I wouldn’t have a student serve as the scribe for another student with learning disabilities unless an IEP specifically directed me to do so.
    - I would have Student Services scribe for a student with a physical disability (like when I had a student with cerebral palsy who is unable to write or draw for himself).

Step 2: Rehearsing evidence without adding more writing
[This usually takes one more day.]

Once you have a student-generated artifact representing students' explanatory models, you can ask them "How do you know this part works this way?" and see if students can connect the "parts" of their story to some specific experiences. Students can add these connections to experience and evidence by placing a sticky note with a short phrase on their story board. When working with students who have typically struggled in school, it is important to pay attention to your tone of voice and social cues when asking questions like "How do you know?" Students can interpret this as a challenge or a put down, which can cause them to shut down or become defensive. Instead, explained to students that you want them to figure out how they learned about parts of their explanation and then pose your question.

Step 3: More rehearsal and time to rethink and rewrite
[This usually takes one more day.]

After each student has worked out their ideas, then students should be ready to communicate with a partner or very small group. However, it's probably not safe to try to have vulnerable students try to communicate in front of a whole group yet, so don't skip this step. Pair-sharing or very small group sharing time allows students to do two things: (1) rehearse their own ideas and language with a small audience and (2) grab ideas and language from their peers as well. You should build in some time for students to add to their storyboards, to delete or change their ideas after hearing from their peers.

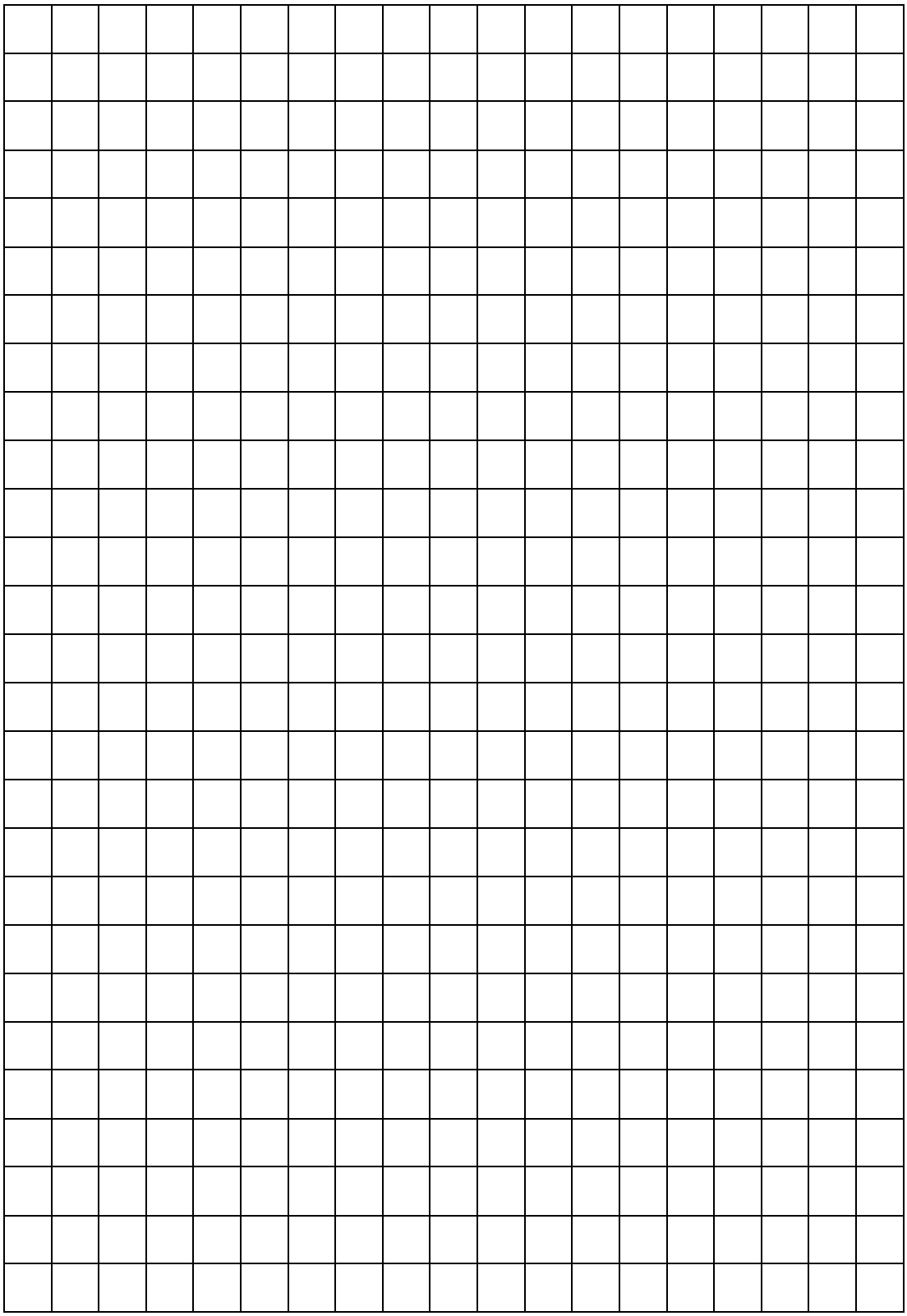
Step 4: The big performance—talking in front of the whole class and the teacher

Now students might be ready to engage in the official dialogue as outlined in this steps of discourse tool 3. If you have done the prep work described here, students will have already formulated and rehearsed their ideas, gathered their thoughts, and tried out some language. When they are now asked to say things out loud (to you or to the whole group) it's not about trying to think on their feet or speak off the top of their head, it's about telling a story that has been rehearsed a little bit over the previous days of class time. Timeframes for all of these steps will condense of it as students become more proficient and comfortable over the year.

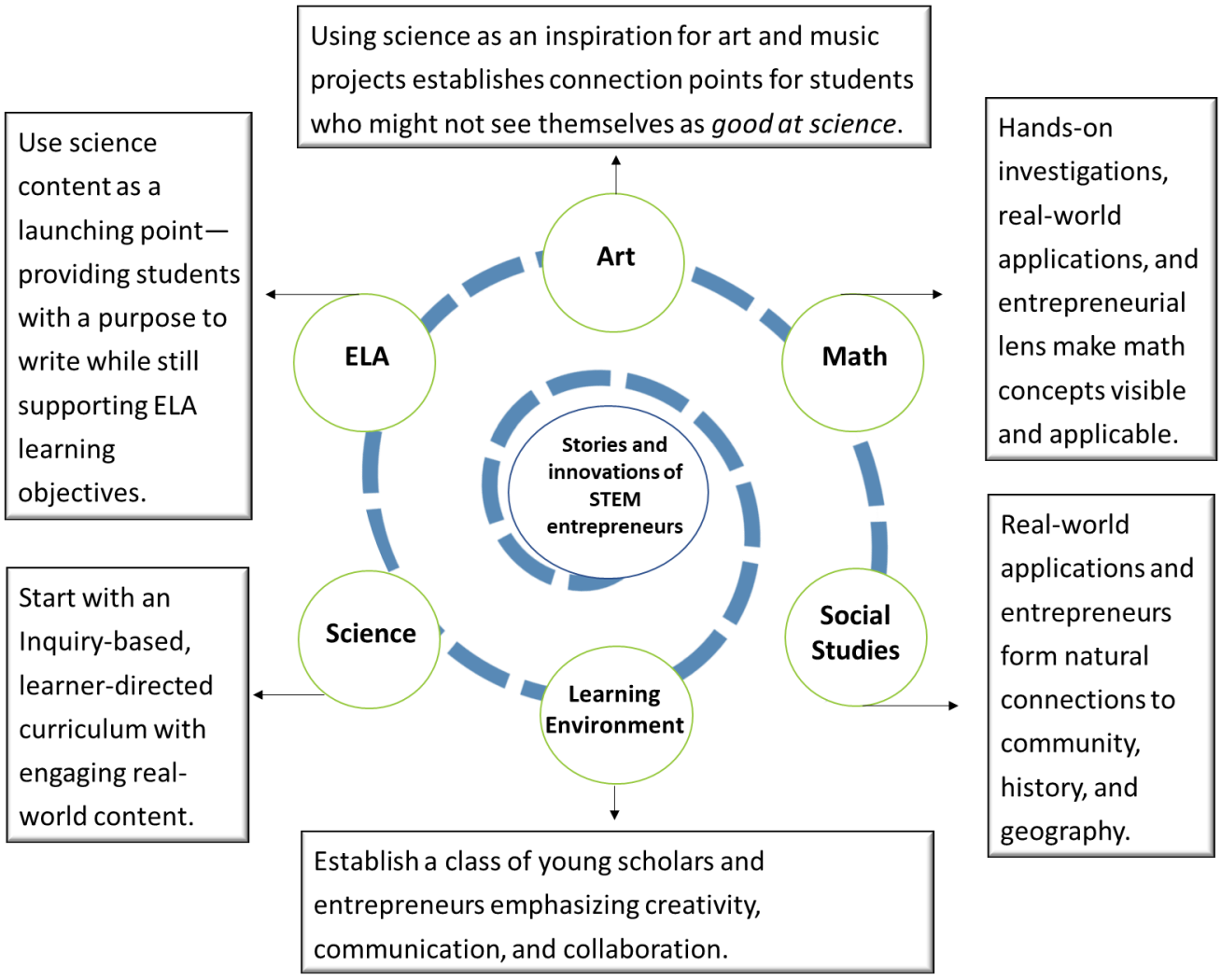
Step 5: The final product—contstructing a polished version of an evidence-based explanation

After all of this, you could ask students to polish up their work from the previous week and write or draw a final draft of their evidence-based explanation.

I often work with colleagues from literacy and language classes to develop science writing, paragraph writing, and essay writing support that is consistent across students' subject areas. Many schools use a writing model to assist students when writing paragraphs and essays.



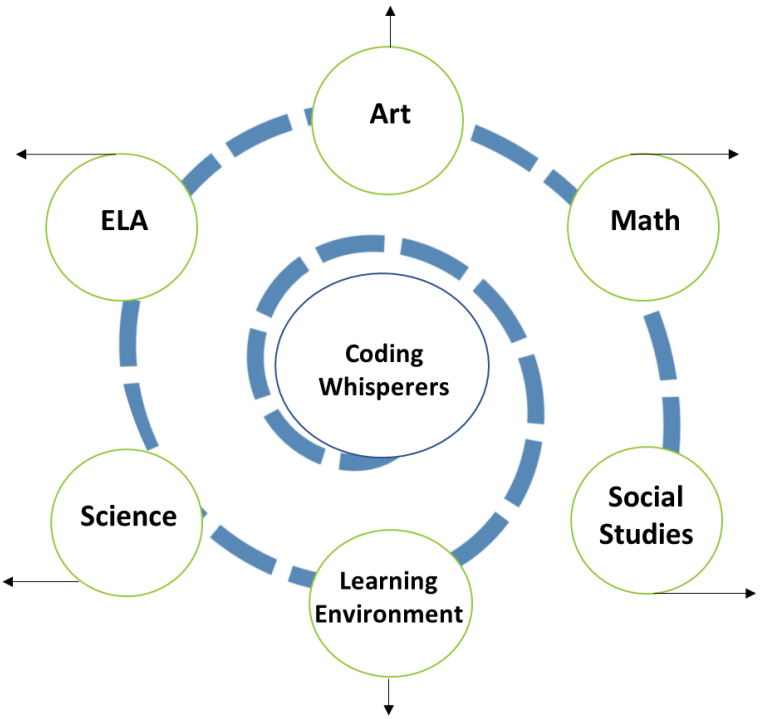
Cross-curriuluar Integration Template (Teaching from the Inside Out)



How can art, music, and sound improve computer programs? (Think about your animations and video games.) How can computer programming improve art and music.

How is coding like a language?

What are some similarities between writing a story and writing a computer program.



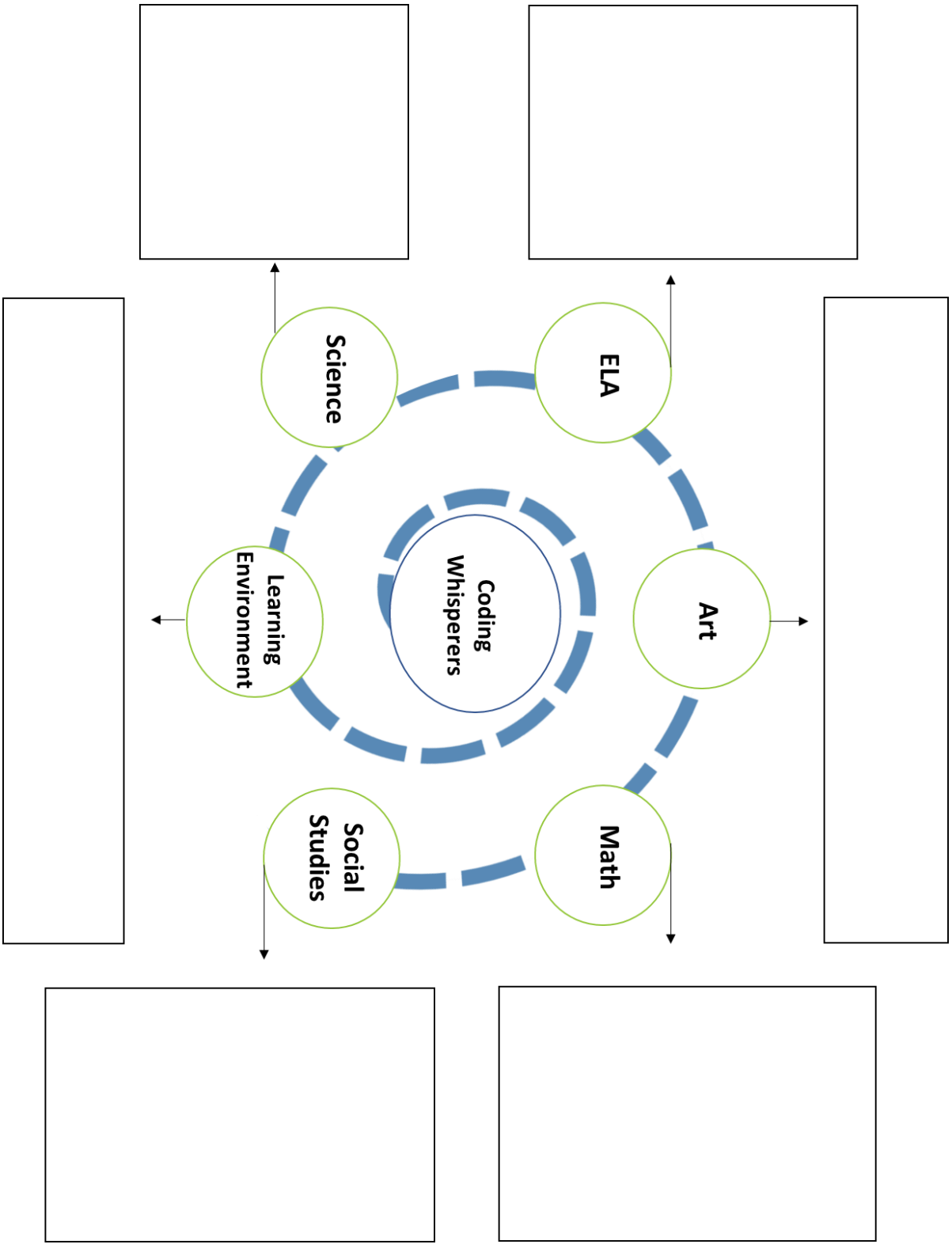
Computer programs rely on math to determine what to do next. Some examples include:

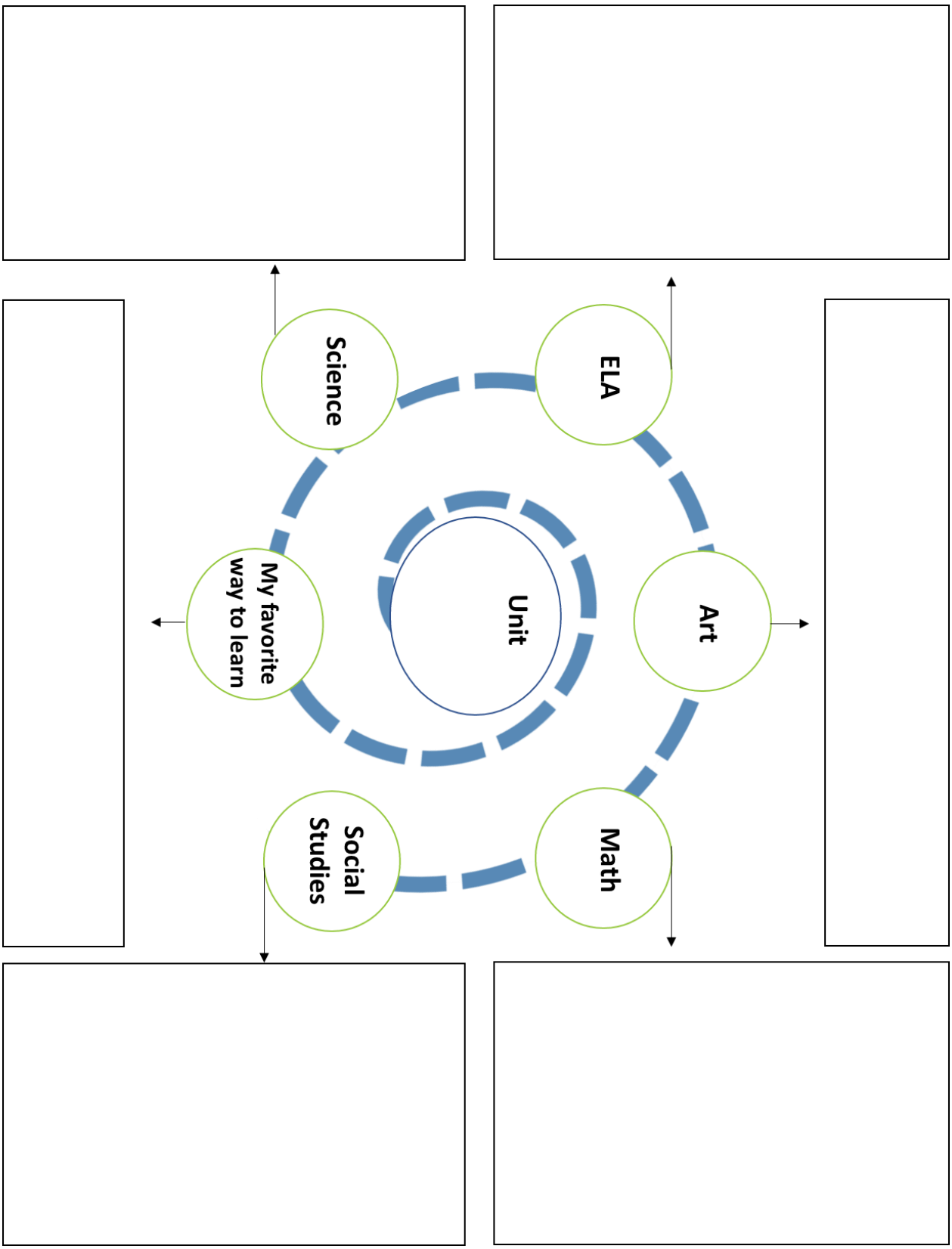
- How long to wait between commands?
- How many repeats of a command?
- How to adjust or move a character.

What is coding? Why is it important? What are some specific skills you need to develop to be a computer programmer?

How has our understanding of numbers, computers, and programming grown over time? How are people using coding and computer programming to improve things in their community? Can you think of a way you can use coding to solve a problem?

How do you design a video game? How can you use coding to solve a problem in my life or community?
How do YOU think we can answer or investigate those questions?





Tips for getting student-directed conversations back on topic

While healthy student-directed discussion is engaging, fun, and informative for students and teachers alike, they can sometimes lead us down a rabbit hole, prompting teachers to ask: "How do I get us back on task?" There is no right or wrong way to do this, but below we list some ideas that you may find helpful.

1. Use the unit phenomenon or theme.

In each unit the investigations, lessons, and discussion prompts all connect to the overarching phenomenon or theme. By asking them directly how the discussion connects back to either of these gives them, or you, a chance to get the conversation back on track. While the specific way you do this will depend on your unique discussion, some ideas are listed below:

- How does this discussion help us figure out how to create a video game?
- What does this discussion teach us about the different ways we interact with computers?

2. Use one of the essential questions.

The essential questions and enduring understandings associated with each unit are another option for getting conversations back on track. They are generally a little more focused on the specific investigation you are discussing than the overarching phenomenon or theme, so may provide slightly more guidance for your students as they work the discussion back to the task at hand. Some ideas are listed below.

- How does this discussion help us think in the way necessary to write a computer program that does what we want it to do?

3. Use one of the standards or objectives connected with the specific investigation are working on.

Finally, a specific standard or stated objective associated with the investigation—these can be found in the Lesson Planning Tools section and are further detailed in the Education Standards section—are a very direct method to focus your students onto the key take-aways from the discussion.

- How does this investigation help us see energy being transferred from place to place?
- How does this design project help us to solve a problem?

Lesson Planning

The remaining pages are provided for you to use for lesson planning.

Preparation:	
Day/Week	Class Activities

