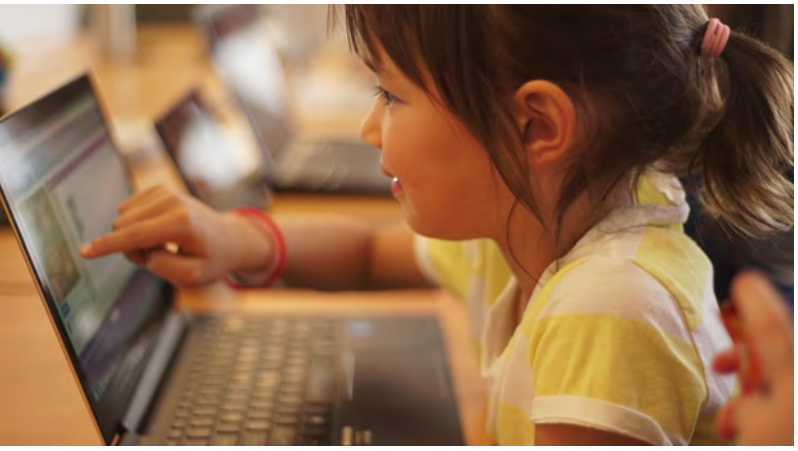
Curriculum Guide
and
Unplugged Lesson Plans

Computer Science Fundamentals

Courses A-F



Welcome to Computer Science Fundamentals! This guide has been created to help you navigate the lessons in Courses A-F. It begins with a brief overview of our entire Computer Science Fundamentals curriculum, followed by tips and tricks for teaching computer science in the classroom. After this valuable information, you will find plans and resources that have been created to make it as easy as possible to run lessons for our newest offerings, Courses A-F.

Are you ready to put the FUN in Fundamentals?

Curriculum Introduction

Who Made This?

Launched in 2013, Code.org® is a non-profit organization dedicated to expanding access to computer science, and increasing participation by women and underrepresented students of color. Our vision is that every student in every school should have the opportunity to learn computer science. We believe computer science should be part of core curriculum, alongside other courses such as biology, chemistry, or algebra.

Code.org increases diversity in computer science by reaching students of all backgrounds where they are — at their skill-level, in their schools, and in ways that inspire them to keep learning. Read about our efforts to increase diversity in computer science [https://code.org/diversity].

At Code.org, we believe in teamwork. That's why we've partnered up with some of the most innovative elementary computer science educators in the country. Our kick-off crew included Alana Aaron and Lionel Bergeron from the New York City Department of Education, as well as Bryan Twarek from the San Francisco Unified School District, Grant Smith of Emerald Data Solutions, Joel Spencer from the Little Rock School District, and Michael Harvey from Falmouth Elementary School...and that was only the beginning!

Over the course of one year, we successfully designed, implemented, and piloted six modified courses. Our pilot was opened to our dedicated Computer Science Fundamentals Facilitators, as well as thousands of engaged educators from around the world. With their help, we were able to create this course in a way that benefits schools of all different shapes and sizes.

Computer Science Fundamentals courses continue to include lessons on Internet Safety and Digital Citizenship, thanks to our friends at Common Sense Education.

As always, it is thanks to our generous donors that we were able to develop and offer this curriculum at no cost to schools, teachers, or students: Microsoft, Infosys Foundation USA, Facebook, Omidyar Network, Google, Ballmer Family Giving, Ali and Hadi Partovi, Bill Gates, The Bill and Melinda Gates Foundation, BlackRock, Jeff Bezos, John and Ann Doerr, Mark Zuckerberg and Priscilla Chan, Quadrivium Foundation, Amazon Web Services, The Marie-Josee and Henry R. Kravis Foundation, Reid Hoffman, Drew Houston, Salesforce, Sean N. Parker Foundation, Smang Family Foundation, Verizon.

Who is This For?

Computer Science Fundamentals was built with elementary school educators in mind. Courses A-F have been specifically tailored to students in Kindergarten through 5th grade, and no prior experience is assumed.

The lessons in CS Fundamentals are presented with the understanding that many teachers will not have any previous computer science training, and educators are therefore encouraged to learn along with their students.

Which Course is For Me?

Computer Science Fundamentals has three different segments, each serving a different purpose:

- Courses A-F
- Courses 1-4
- The Accelerated Course

Computer Science Fundamentals, Courses A-F

These courses are the new preferred path for Computer Science Fundamentals. The courses are roughly aligned to each year of elementary school, allowing for the most robust content along the entire elementary pipeline.

Course A	Course B	Course C	Course D	Course E	Course F
Designed for Kindergarten	*Designed for 1st Grade*	*Designed for 2nd Grade*	*Designed for 3rd Grade*	*Designed for 4th Grade*	*Designed for 5th Grade*

Ideal entry points exist at kindergarten, first, and second grades. Suitable entry points have been created in courses D, E, and F with the addition of age-appropriate introduction lessons that present content from previous courses at an accelerated pace.

The core content of these courses range from approximately 12 lessons in A & B, to nearly 20 lessons in Course F, with additional lessons available on the Code.org website to further support specific concepts.

Computer Science Fundamentals, Courses 1-4

Originally created for beginners of any age, Courses 1-4 were intended for classrooms where students vary widely in age and or ability. In the 2017-2018 school year, Courses A-F are still being translated into multiple languages. If your students need to work in Spanish, Turkish, or other languages, Courses 1-4 are the best choice this year. These courses are no longer being updated, but they will continue to be supported into the foreseeable future.

Course 1	Course 2	Course 3	Course 4
Beginning for early readers	*Beginning for comfortable readers*	*Sequel to Course 2*	*Sequel to Course 3*

This series has only two entry points: Course 1 for early readers and Course 2 for readers who can comfortably sound out words. Each of the latter courses assume complete knowledge of the courses that come before them.

The core content of these units include roughly 18-20 lessons per course, with additional lessons available on the Code.org website to further support specific concepts.

Since these courses are being deprecated we do not suggest embarking on this pathway if you have not already accumulated hours with them, unless you require materials in languages other than English. Those who currently use them in classrooms can continue to do so without fear that they will disappear in the coming years.

Computer Science Fundamentals, Express

Intended for students ages 10 and older, the Express Course is a powerful set of puzzles that introduces the best of computer science and computational thinking in one continuous burst. This course combines the concepts in Courses A-F, in one accelerated series. This works well for older students (middle or high school) that didn't go through the CS Fundamentals courses in elementary school.

It's also a great choice if your elementary school curriculum only has time for one course and you're working with older students (4th or 5th grade). Teachers also use this for students who are already comfortable with CS concepts, or if they pick up on ideas quickly.

CS Fundamentals - Express
One quick course for older students who did not experience CS Fundamentals in elementary school

This course will grow alongside courses A-F, containing the most effective favorites from that series. The length of this course may fluctuate between 25 and 35 lessons as it evolves.

Course Structure

Each Computer Science Fundamentals course is a little bit different, but they all value the method of introducing concepts with unplugged activities before bringing students to an online workspace.

Conceptual Chunks

The unplugged lessons and online puzzle tutorials are chunked together by shared concepts, where the unplugged lesson serves as a fun and gentle introduction to a computing concept that is further explored through coding exercises. This allows each “chunk” to be separated into groups of lessons that can be taught within a defined time period or as a sub-unit.

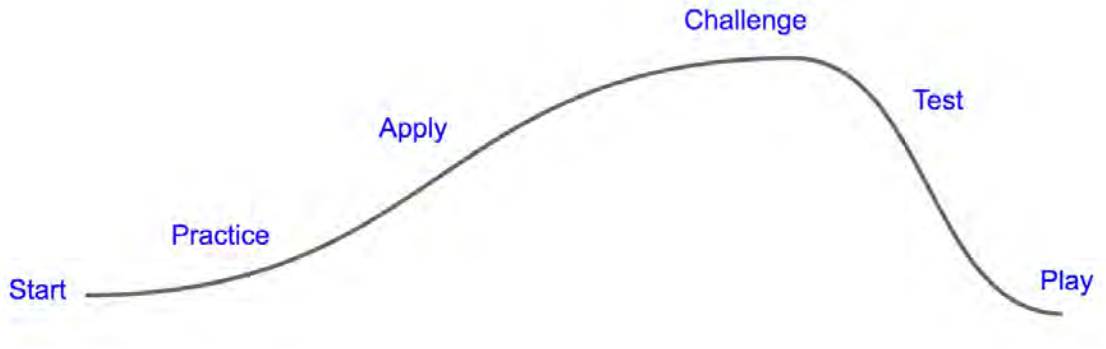
See this example from Course C in which the unplugged lesson, Getting Loopy, precedes three online lessons.

A selection of four lessons from CS Fundamentals, Course C

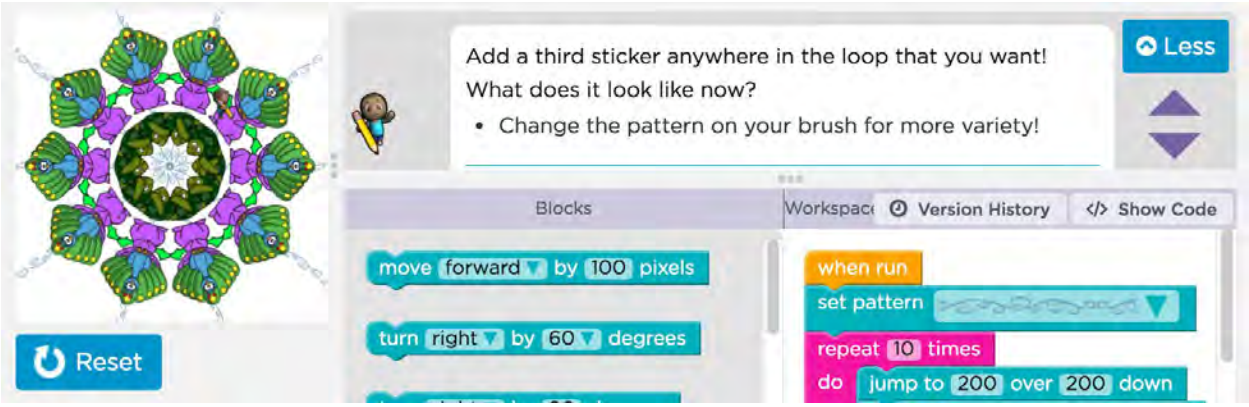
Lesson Structure

In addition, Courses A-F have an overarching lesson architecture where concepts are carefully introduced using a structure that has been tailored to set classrooms up for success. These puzzle progressions start by introducing ideas in a step-by-step manner, then proceed to a brain-bending challenge before setting students gently back into a puzzle that perfectly represents the level of understanding that was intended for that series.

See the diagram below to get a better idea of how complexity changes over the course of an online lesson.



Several concepts also include additional lessons that focus less on skill building and focus more on creative problem solving. As an example, these lessons might include puzzles with less obvious pathways, a complex use of concepts, or multiple correct solutions.



End of Course Projects (Courses E & F)

While each course offers the opportunity for students to take what they've learned at the end of a lesson and put it together into a unique project that represents their own creativity, Courses E & F take student projects to a whole new level.

In the final two courses in the A-F series, project development takes the stage. Here, students are encouraged to plan, build, revise, and present projects of their own. Following a project from inception to delivery offers an inside look at the software development cycle. This guided project offers scaffolded rubrics for the benefit of both student and teacher.

Technology Requirements

A computing device and an Internet connection.

We work hard to build an environment that supports all modern web browsers on desktops and mobile devices. This includes Internet Explorer 11+ and the latest versions of Firefox, Chrome, Safari.

Our instructional videos may be affected depending on your school's internet filters. If YouTube is blocked at your school, our video player will attempt to use our non-YouTube player instead. For more details about the IT requirements for accessing and playing our embedded videos, see our IT requirements page at https://code.org/educate/it. We've made all of our videos available for download using a link located in the bottom corner. If all fails, some videos have a "Show Notes" tab that provide a storyboard equivalent of the video.

Implementation Tips and Considerations

This document offers suggestions for implementing an individual CS Fundamentals course in an elementary school classroom, as well as planning the rollout of all Courses A-F as a pathway across elementary school grade levels.

Scheduling The Lessons

Where do you teach Computer Science Fundamentals? Do you use an existing "special" time like for media (library) or art, or do you use a pre-existing computer lab period and split the online and unplugged lessons between the lab teacher and the grade level teachers respectively? Whether you are a grade level teacher or a specialist (media, art, technology), the CS Fundamentals courses are designed to be flexibly implemented.

While each course is setup to run as one lesson per week for a semester, they can alternatively be run two or three times a week for about two months, or one lesson every two weeks for a year. Lessons are meant to be completed in order and can range from as little as 20 minutes to more than 45 minutes if extension activities are included.

Teaching in the Classroom

Here are implementation tips for four common situations in elementary school:

- **Grade Specific Classroom**
  - Even if you have 1:1 computers, consider grouping students up for pair programming. The benefit of learning to communicate problems and solutions to a peer far outweigh the risks of only having one student at the keyboard at a time.
  - You do not need computers for unplugged days, unless you plan to double up and work on an online lesson immediately after the offline one. Even still, consider giving time for the concept to sink in after an unplugged lesson before moving to online programming.

- Students should be encouraged to travel from concept to concept with one another, but should not be made to keep in lockstep puzzle-by-puzzle. Make sure students feel empowered to travel at their own pace.
- End of Lesson puzzles should be seen as a great opportunity to showcase the things students have learned during the day, not as a required assessment for the segment.

- **Mixed Grade Classroom**
  - When selecting groups for pair programming, make sure that students are paired either with students of a similar age or a similar experience level. Having pairs that are too far separated in skills is not fair to either member of the team.
  - If your class is regularly held in the lab, look for rooms to "visit" for unplugged activities (like the library or the gym.) This will give students room to spread out and feel like they are learning authentically, rather than trying to "make do" in the available space.
  - It's expected that students from different grades could be on different lessons or even in different courses. Decide on what rules you want to foster surrounding these differences and get buy-in from students before each lesson to prevent things from getting unruly.

- **Using "Centers" or "Stations"**
  - Some classrooms have a small number of computers set up in one area and teachers use these as activity centers. This can be very effective once the class has already gone through the unplugged activities for a concept together.
  - Continue to encourage pair programming, even if students are only able to get through 2 to 4 puzzles at a time. The realizations that they have while learning to talk through problems will far outweigh any reduction in speed that they experience while trading off.
  - If you have empty stations open, offer unplugged activities that relate back to the online lessons. Providing extra opportunities for context switching will help to solidify ideas that might otherwise cause them to struggle.

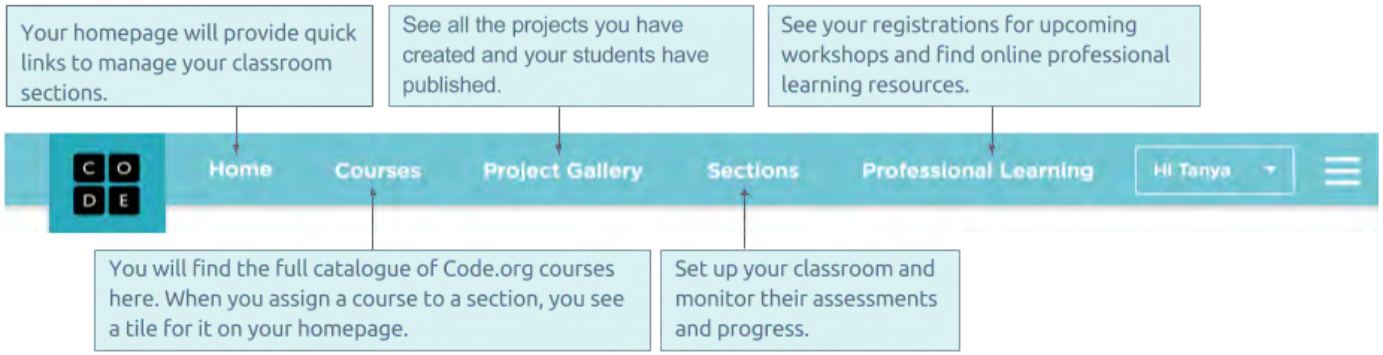
- **Every Now and Again Lessons**
  - If you are a teacher that struggles to find a time and place to integrate computer science into your annual curriculum, you are in good company. Remember, when taught well, any computer science is better than no computer science.
  - Choose a concept and teach it thoroughly.
    - In elementary school, the main goal is to teach students that they are capable of learning computer science. If you ditch a deep dive on concepts in favor of a shallow introduction, students might be left feeling as if they don't understand any of it.
    - Start with the unplugged lesson for the concept that you are choosing. Use the word (such as "loops") often in class for days after the unplugged lesson has been completed.
    - On your next trip to the computer lab, start by completing a bridging activity that ties your concept to the online lessons that students are about to do. Spend that day in the lab doing the straightforward, educational puzzles that come in the lesson directly following the unplugged activity in the course.

- If you have another chance to get to the lab, explore other online puzzles with that same concept. This can be either a second concept lesson on Code.org, or a supporting lesson from another interface (such as Scratch or the Foos.)

Navigating the Code.org Website

Code.org Site Navigation

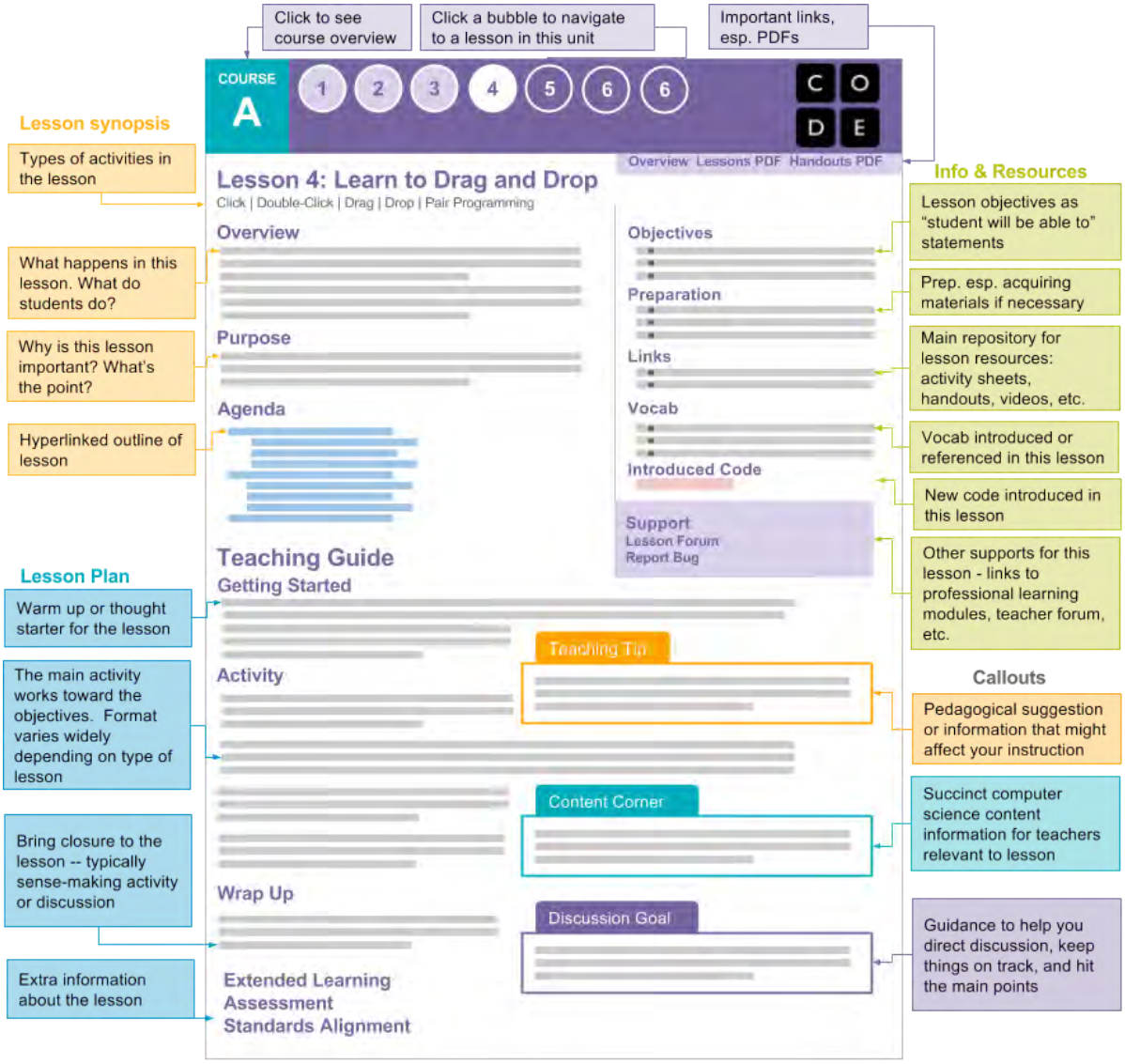
Log into Code.org with your teacher account. The website header will help you navigate the site:



Once you've assigned CS Fundamentals to your section, click the tile on your homepage to reach the course overview page. This is your starting point for lesson planning, professional learning, and all the resources you need to teach the course.

The Code.org Lesson Plan Structure

Every lesson plan has a common structure that should make it easy to find what you need. Planning for a lesson starts by looking at the overview, then reviewing the core activity to get a deeper sense of what it is and how long it might take.

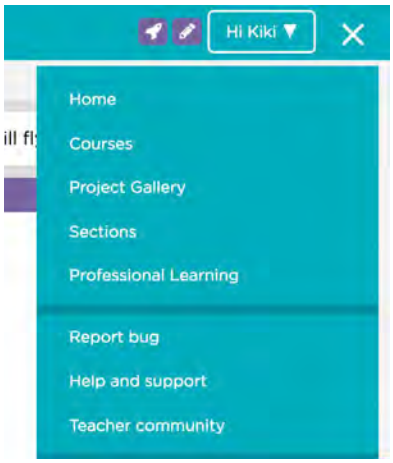


Lesson Length

Lessons in CS Fundamentals are written for a wide variety of classrooms. We generally try to make one lesson equal to one 30 to 45 minute class period. However some lessons take multiple days, such as projects or concepts that do not easily break down into separate lesson plans. Many lessons include time estimates, but these vary based on the age of your students, their background with the material, and their interests.

Getting Help

The curriculum is completely free for anyone to teach anywhere. For support, click on the menu in the upper right-hand corner of the website.



Here, you'll find our "Help and support" forum where you can email us or find how-to articles. You'll also see a link to our "Teacher community" forums where you can connect to other teachers for support, teaching tips, or best practices.

When you're in a puzzle, you'll see an additional "Report bug" link for that puzzle. Thank you for helping us find and fix any issues!

If you are a teacher and you'd like to attend a free training on our K-5 Computer Science curriculum, you can find links to local workshops by visiting https://code.org/professional-development-workshops.

Curriculum Values

While Code.org offers a wide range of curricular materials across a wide range of ages, the following values permeate and drive the creation of every lesson we write.

Computer Science is Foundational for Every Student

We believe that computing is so fundamental to understanding and participating in society that it is valuable for every student to learn as part of a modern education. We see computer science as a liberal art, a subject that provides students with a critical lens for interpreting the world around them. Computer science prepares all students to be active and informed contributors to our increasingly technological society whether they pursue careers in technology or not. Computer science can be life-changing, not just skill training.

Teachers in Classrooms

We believe students learn best with the help of an empowered teacher. We design our materials for a classroom setting and provide teachers robust supports that enable them to understand and perform their critical role in supporting student learning. Because teachers know their students best, we empower them to make choices within the curriculum, even as we recommend and support a variety of pedagogical approaches. Knowing that many of our teachers are new to computer science themselves, our resources and strategies specifically target their needs.

Student Engagement and Learning

We believe that students learn best when they are intrinsically motivated. We prioritize learning experiences that are active, relevant to students' lives, and provide students authentic choice. We encourage students to be curious, solve personally relevant problems and to express themselves through creation. Learning is an inherently social activity, so we interweave lessons with discussions, presentations, peer feedback, and shared reflections. As students proceed through our pathway, we increasingly shift responsibility to students to formulate their own questions, develop their own solutions, and critique their own work.

Equity

We believe that acknowledging and shining a light on the historical inequities within the field of computer science is critical to reaching our goal of bringing computer science to all students. We provide tools and strategies to help teachers understand and address well-known equity gaps within the field. We recognize that some students and classrooms need more supports than others, and so those with the greatest needs should be prioritized. All students can succeed in computer science when given the right supports and opportunities,

regardless of prior knowledge or privilege. We actively seek to eliminate and discredit stereotypes that plague computer science and lead to attrition of the very students we aim to reach.

Curriculum as a Service

We believe that curriculum is a service, not just a product. Along with producing high quality materials, we seek to build and nourish communities of teachers by providing support and channels for communication and feedback. Our products and materials are not static entities, but a living and breathing body of work that is responsive to feedback and changing conditions. To ensure ubiquitous access to our curriculum and tools, they are web-based and cross-platform, and will forever be free to use and openly licensed under a Creative Commons license.

Pedagogical Approach To Our Values

When we design learning experiences, we draw from a variety of teaching and learning strategies all with the goal of constructing an equitable and engaging learning environment.

Role of the Teacher

We design curriculum with the idea that the instructor will act as the lead learner. As the lead learner, the role of the teacher shifts from being the source of knowledge to being a leader in seeking knowledge. The lead learner's mantra is: "I may not know the answer, but I know that together we can figure it out." A very practical residue of this is that we never ask a teacher to lecture or offer the first explanation of a CS concept. We want the class activity to do the work of exposing the concept to students, allowing the teacher to shape meaning from what the students have experienced. We also expect teachers to act as the curator of materials. Finally, we include an abundance of materials and teaching strategies in our curricula - too many to use at once - with the expectation that teachers have the professional expertise to determine how to best conduct an engaging and relevant class for their own students.

Discovery and Inquiry

We take great care to design learning experiences in which students have an active and equal stake in the proceedings. Students are given opportunities to explore concepts and build their own understandings through a variety of physical activities and online lessons. These activities form a set of common lived experiences that connect students (and the teacher) to the course content and to each other. The goal is to develop a common foundation upon which all students in the class can construct their understanding of computer science concepts, regardless of prior experience in the discipline.

Materials and Tools

Our materials and tools are specifically created for learners and learning experiences. They focus on foundational concepts that allow them to stand the test of time, and they are designed to support exploration and discovery by those without computer science knowledge. This allows students to develop an understanding of these concepts through "play" and experimentation. From our coding environments to our non-coding tools and videos, all our resources have been engineered to support the lessons in our curriculum, and thus our philosophy about student engagement and learning. In that vein, our videos can be a great tool for sensemaking about CS concepts and provide a resource for students to return to when they want to refresh their knowledge. They are packed with information and "star" a diverse cast of presenters and CS role models.

Creation and Personal Expression

Many of the projects, assignments, and activities in our curriculum ask students to be creative, to express themselves, and then to share their creations with others. While certain lessons focus on learning and practicing new skills, our goal is always to enable students to transfer these skills to creations of their own. Everyone seeks to make their mark on society, including our students, and we want to give them the tools they need to do so. When computer science provides an outlet for personal expression and creativity, students are intrinsically motivated to deepen the understandings that will allow them to express their views and carve out their place in the world.

The Classroom Community

Our lessons almost always call for students to interact with other students in the class in some way. Whether learners are simply conferring with a partner during a warm up discussion, or engaging in a long-term group project, our belief is that a classroom where students are communicating, solving problems, and creating things is a classroom that not only leads to active and better learning for students, but also leads to a more inclusive classroom culture in which all students share ideas and listen to ideas of others. For example, classroom discussions usually follow a Think-Pair-Share pattern; we ask students to write computer code in pairs; and we strive to include projects for teams in which everyone must play a critical role.

Classroom Practices

The classroom practices for CS Fundamentals are different strategies that are used repeatedly in many different lessons and units. These six classroom practices are at the core of the ways the curriculum is designed as we believe these are best practices that lead to positive classroom culture and ultimately student learning.

The 6 Main Classroom Practices of CS Fundamentals:

- Lead Learner
- Pair Programming
- Authentic Choice
- Unplugged Activities
- Bridging Activities
- Journaling

Lead Learner

What is it?

The curriculum has been written with the idea that the instructor will act as the lead learner. As the lead learner your role shifts from being the source of knowledge to being a leader in seeking knowledge. The lead learner's mantra is: "I may not know the answer, but I know that together we can figure it out." The philosophy of the lead learner is that you don't have to be an expert on everything; you can start teaching CS Fundamentals knowing what you already know, and learn alongside your students. To be successful with this style of teaching and learning, the most important things are modeling and teaching how to learn.

How does it connect to the curriculum?

One of the Code.org curriculum values is developing teachers who are new to computer science. In order to support those teachers, the curriculum is set up to create an engaging and relevant class that helps students uncover and develop the knowledge they need. This makes it possible for a teacher to lead the course without knowing all of the answers at first, as long as they embrace the lead learner role. In addition, it is not possible

to have complete command over every rapidly-changing facet of computer science, no matter how much experience you have. Rather than feeling daunted, the lead learner welcomes this fact.

We believe that the lead learner technique represents good teaching practice in general. Acting as the lead learner is an act of empathy toward your students and the challenges they face in learning material for the first time. One important job of the teacher in the CS Fundamentals classroom is to model excitement about investigating how things work by asking motivating questions about why things work the way they do, or why they are the way they are. With your guidance, students will learn how to hypothesize; ask questions of peers; test, evaluate, and refine solutions collaboratively.

How do I use it?

- Allow students to dive into an activity without presenting all the content first
- Encourage students to rely on each other for support
- Don't give the answer right away, even if you know it
- Feel open to making mistakes in front of students so that they see it is part of the learning process
- Ask students questions that direct their attention toward the issue to investigate without giving away what they need to change
- Model the steps you would go through as a learner of a new subject. Explain the different questions you ask yourself along the way and the ways you go about finding answers

Pair Programming

What is it?

Pair programming is a technique in which two programmers work together at one computer. The "driver" writes code while the "navigator" directs the design and setup of the code. The two programmers switch roles often. Pair programming has been shown to:

- improve computer science enrollment, retention, and students' performance
- increase students' confidence
- develop students' critical thinking skills
- introduce students to a "real world" working environment

How does it connect to the curriculum?

In CS Fundamentals there are many lessons on the computer (plugged lessons) during which students develop programming skills through online progressions. Pair programming can help to foster a sense of camaraderie and collaboration in your classroom during sets of plugged lessons. It has been shown to increase the enrollment, retention, and performance of students in computer science classes. It promotes diversity in the classroom by reducing the the so-called "confidence gap" between female and male students, while increasing the programming confidence of all students.

How do I use it?

To get students pair programming:

1. Form pairs
2. Give each pair one computer to work on
3. Decide upon initial roles
4. Have students start working
5. Ensure that students switch roles at regular intervals (every 3 to 5 minutes)
6. Ensure that navigators remain active participants

It can be hard to introduce pair programming after students have worked individually for a while, so we recommend that teachers start with pair programming in the first few plugged lessons. Just like any other classroom technique, you may not want to use this all the time as different types of learners will respond differently to working in this context. Once you have established pair programming as a practice early on, it will be easier to come back to later.

Resources

Code.org also has a feature to help both students get "credit" on their accounts for the work they do together. Check out the blog on Pair Programming: teacherblog.code.org/post/147349807334/try-pair-programmingtrack-the-progress-of

Videos:

- For Teachers: youtu.be/sxToW3ixrwo
- For Students: youtu.be/vgkahOzFH2Q

The National Center for Women & Information Technology (NCWIT) has a great resource about the benefits of pair programming. Check it out at: www.ncwit.org/resources/pair-programming-box-power-collaborative-learning

Authentic Choice

What is it?

Authentic choice is the practice of allowing students to decide on the focus of their creation when they are working on a project. This can be scoped in different ways with different projects, but the central point is to allow students to work on something they are personally invested in.

How does it connect to the curriculum?

In the curriculum, we give students many opportunities to work on projects that we hope will feel personally relevant. Whether it be a small freeplay level at the end of a lesson, or a course project designed by students in older elementary, every student should get ample opportunity to develop creations of their own.

In addition, we encourage teachers to help students utilize their new skills in creative ways at the end of each lesson, using the Lesson Extras option. There, students will find challenge puzzles and open-canvas projects to use for deeper learning and self-expression.

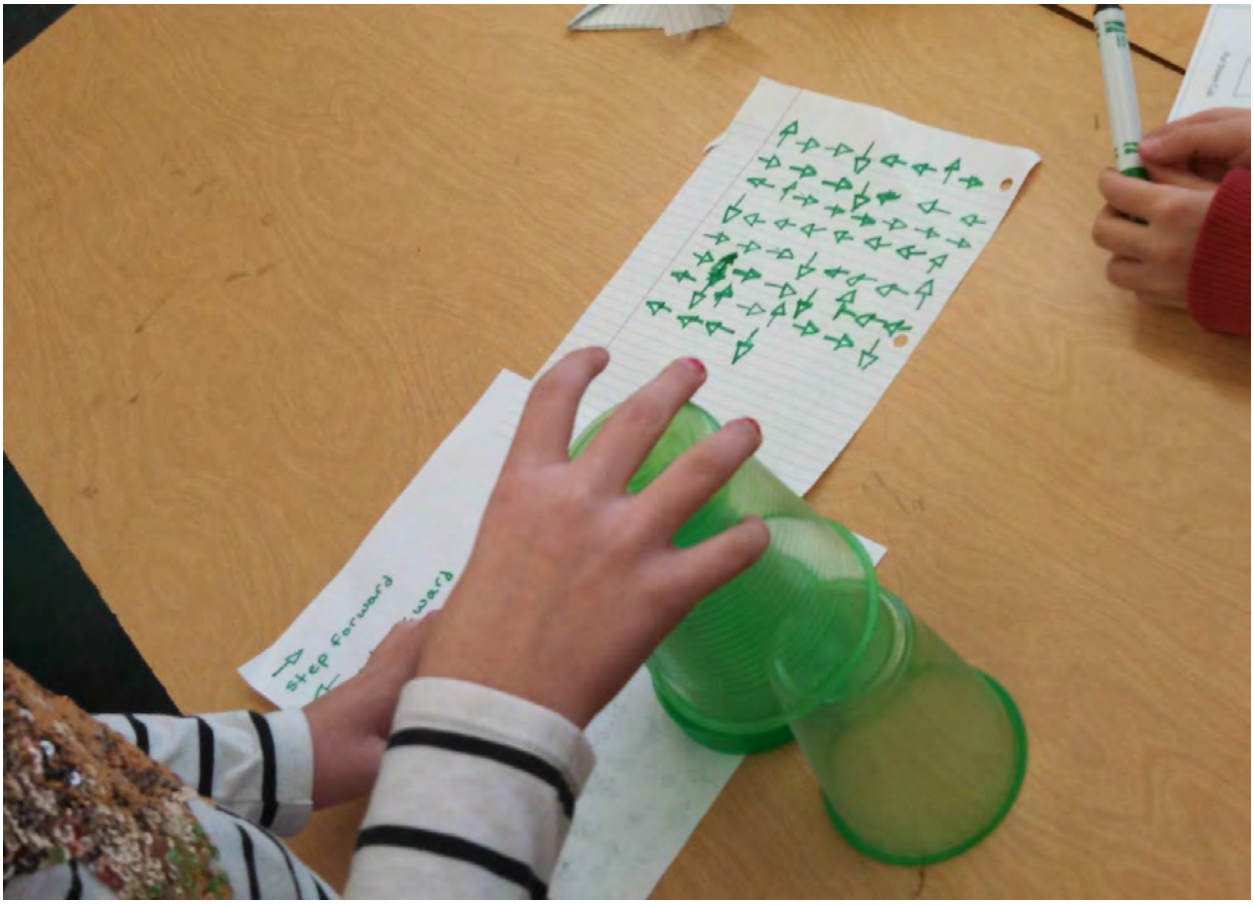
How do I use it?

- Give students time to get creative and find something they are passionate about in the project that they are working on
- Encourage students to find personally relevant contexts for the work they do
- Try to keep the projects as open to students' interests as possible while still keeping them focused on the learning at hand

Unplugged Activities

What are they?

In Code.org curriculum, we refer to activities where students are not on the computer as "unplugged" lessons. Lessons where they are on the computer are called "plugged" lessons.



How do they connect to the curriculum?

Unplugged activities are more than just an alternative for the days when the computer lab is full. They are intentionally-placed kinesthetic opportunities that help students digest complicated concepts in ways that relate to their own lives. When we write the curriculum, we plan the progression of unplugged and plugged lessons to build on each other. Often something that is done in an unplugged environment is setting the stage for or reviewing a concept done in a plugged environment. Both are vital pieces of the curriculum as they build

student knowledge and understanding in different ways. In addition, unplugged lessons can help build and maintain a collaborative environment in your classroom. An unplugged environment also can be a good way to check for student understanding.

How do I use them?

- Don't skip these lessons! These lessons often involve more advance preparation but are worth it!
- Try to present the lessons in the order they appear as best as possible
- Help students make sense of how the activity connects to the concepts they are learning in earlier and later lessons with "bridging activities"
- Call back to relevant unplugged activities during plugged lessons
- For a list of all unplugged activities covered in the CS Fundamentals curriculum (plus a few extras!) visit https://code.org/curriculum/unplugged

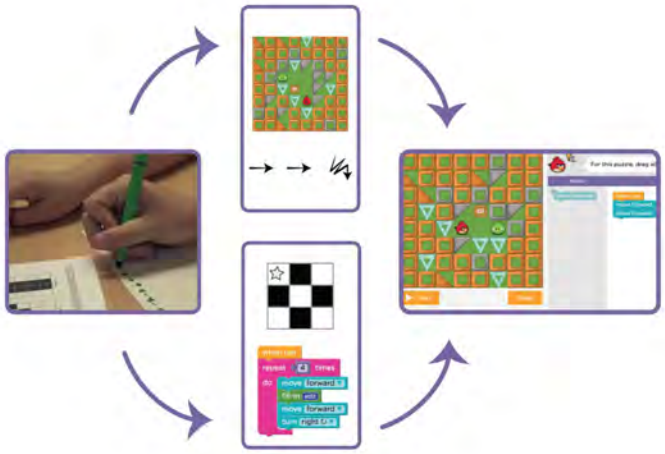
Bridging Activities

What are they?

In the Computer Science Fundamentals curriculum, we refer to activities that mix online and offline elements as "bridging activities."

How do they connect to the curriculum?

Bridging activities connect our unplugged activities to our online lessons in a real and definitive way. They often exist as a method of turning an abstract concept idea learned through play, like conditionals, into an actionable tool for the plugged puzzles.



How do I use them?

- Include a bridging activity after each unplugged lesson, but before the next plugged series (ideas are provided in the “Warm-Up” section of the relevant online lesson plans)
- Use encouraging language to help students transition from the previous concept situation to the programming context
- Feel free to come up with your own online/offline blends to keep the curriculum relevant to your classroom

Journaling

What is it?

In CS Fundamentals, students are encouraged to keep a journal nearby to write down thoughts and answer questions.

How does it connect to the curriculum?

Courses A-F of Computer Science Fundamentals were written with the importance of journaling in mind. Journaling for reflection is a popular tool in education, but we take that one step further. Like a chemist would catalog strategies and solutions, so do we ask our budding computer scientists to take notes on their trials and achievements. Journals are useful as scratch paper for building, debugging, and strategizing, and they offer a fantastic resource for referencing previous answers when struggling with more complex problems.

How do I use it?

- Encourage students to keep their journals beside them at all times when coding
- Remind students that they can write solutions out longhand, then circle patterns to find prime opportunities for loops and functions
- Have students copy down answers to puzzles that they might need in future levels
- Ask students to draw emoticons at the top of the pages to help them identify how they’re feeling about concepts
- End each lesson with a thought or question that students can answer in writing as a way of reflecting on their growth for the day

Student Practices

Students in CS Fundamentals work in a wide array of contexts, but these experiences are tied together by a core set of practices they develop throughout the course. These student practices provide coherence and serve as helpful reminders of the high-level skills and dispositions they should be continually developing.

Problem Solving

- Use a structured problem solving process (see next section) to help address new problems
- View challenges as solvable problems
- Break down larger problems into smaller components

Persistence

- Value and expect mistakes as a natural and productive part of problem solving
- Continue working towards solutions in spite of setbacks
- Iterate and continue to improve partial solutions

Creativity

- Incorporate your own interests or ideas into your work
- Experiment with new ideas and consider multiple possible approaches
- Extend or build upon the ideas and projects of others

Collaboration

- Work with others to develop solutions that incorporate all contributors
- Mediate disagreements and help teammates agree on a common solution
- Actively contribute to the success of group projects

Communication

- Structure your work so that it can be easily understood by others
- Consider the perspective and background of your audience when presenting your work
- Provide and accept constructive feedback in order to improve your work

Skills and Strategies

Tackling Lesson Progressions

Growth Mindset

Throughout these lessons, it is critical that students understand that nobody is born knowing computer science. Everyone starts at the beginning, experiences failure, learns from mistakes, then experiences success. Students should not view frequent programming failures as a sign that they lack talent, but should instead be proud of their own persistence as they make changes and try again. All computer scientists fail. The best programmers are the ones who keep trying long enough to learn from the problems that they encounter. Here are some tips to help your students stay persistent:

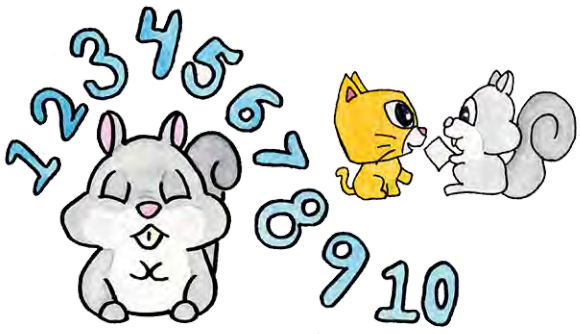
- Keep track of what you have already tried
- Figure out what is happening
- Understand what is supposed to happen
- Look at what that tells you
- Make a change and try again

Frustration

Frustration is a natural part of learning. Instead of trying to avoid frustration at Code.org, we embrace it. The key to persistence is in learning that frustration is a positive sign, not a negative one. When a student gets frustrated, it means that they are about to learn something new. It can be compared to the pain you feel in your mouth before you get a new tooth, or the soreness in your muscles before you get stronger. Congratulate students on recognizing and persevering through frustration.

Here are some tips to help your students deal with frustration:

- Count slowly to 10
- Take some deep breaths
- Write your worries in a journal
- Talk to a partner about your feelings
- Ask for help



Paper and Pencil

It is common for a teacher to expect students to work out code in their head. This puts unfair stress on students, as programs can get complicated and confusing. Encourage students to code with a piece of paper and pencil nearby so that they can scribble down predictions and trace out paths for solving puzzles. Paper can be helpful to record the ideas that have already been tried, as well as thoughts about what can be tried next. It might even help students to write a little about what they did to pass a level so that they can share with classmates who are in need of a little help.

Trying (Then Skipping) Challenges

When Code.org places so much value on persistence, it is hard to justify skipping puzzles because they feel too difficult or confusing. It is, however, how our challenge puzzles were designed.

Most of our online lessons contain one “challenge” puzzle near the end of the series. This puzzle was carefully placed to inspire students to practice new concepts in a different way. It will test their persistence, highlight misconceptions, and hopefully lead them to the “ah-ha” moment that educators love! Additionally, spending a fair amount of time with the challenge puzzle will help the following levels feel simple in comparison, even though the levels that follow demonstrate the exact target of understanding expected from a student by the time they finish each lesson.

In order to provoke those strong accomplishments, challenge levels must all but guarantee that they cannot be solved quickly. Because of this, they will be far too hard for many students, which means that the messaging of these challenges is extremely important. Keep these things in mind, and make sure that you review them with your students before each visit to the computer lab.

- When you get to the challenge puzzle, give it your best try!
- Keep notes as you try to solve the puzzle so that you don’t lose your place or try the same thing over and over again.
- These puzzles are supposed to be hard. You don’t have to solve it in order to learn from it, but you do have to try your best!
- Remember that you can come back to the challenge puzzle later. If you start to get frustrated, go ahead and skip the challenge until you complete the rest of the lesson. You can come back to it when you’ve finished everything else!

Lesson Extras

Many teachers like to keep their entire class on the same lesson instead of letting students skip ahead to new concepts. For that reason, we have provided an area at the end of each lesson series where students can use the concepts that they’ve learned in new and interesting ways. These puzzles are considered “optional.” They are a sandbox to keep the quickest students challenged and entertained until the end of class. In addition, the

Lesson Extras can be used to add an additional day onto concept lessons, offering the opportunity for students to come back and create personal and authentic projects with their newest concepts.

Copy/Paste

Though we do not make it obvious, most individual blocks in CS Fundamentals are able to be copied/pasted within the same level. If you feel that the ability to copy and paste blocks will enhance your student's learning experience, then you can share the copy/paste method with them:

1. Highlight the block that you want to copy
2. Press ctrl+c to copy
3. Press ctrl+v to paste
4. Move your pasted block to where you would like it

While this "hack" generally only allows you to copy one block at a time, there is a work around for duplicating larger chunks of code. All you need to do is pull a statement block (like a loop, conditional, or simple function) from the toolbox and place all of the code to be copied inside. Now, select the statement block and copy/paste. All of the code inside will be duplicated. You can now pull the extra code out of the statement block and use it as desired.

Puzzle/Problem Solving Recipe

Start a Puzzle with Understanding

Often, the quickest way to pass a level is to know where you're headed from the start. Help students understand that being successful means taking a look at what is required before dragging anything out to the workspace. Here are some important questions that students should ask:

- What do the instructions say?
- What am I supposed to do?
- What has already been done?
- Are there any questions that I need answers to?
- Have I solved another problem like this one?
- Can I put the puzzle's goal into my own words?

Debugging

Small misunderstandings can lead to big problems! Help keep students moving by showing them some basic debugging skills.

Step Button

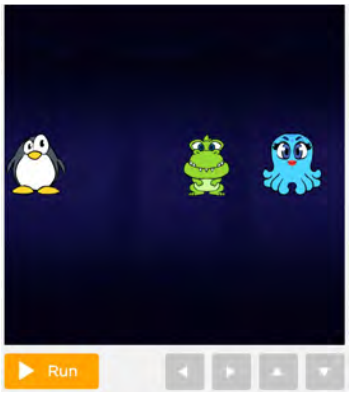
Most puzzles in CS Fundamentals have a "Step" button (excluding Artist, Play Lab, and BB-8 levels.) With the "Step" button, it is possible to go through a program block by block to see what happens each step of the way. This is a helpful tool when your code moves too quickly to understand where things get off course.

To use the “Step” button, simply click on “Step” instead of “Run”. Your code will run exactly one block before coming to a rest again. To continue through the code, keep pressing “Step” until you have completed your program, or found your bug!

In Artist puzzles, the “Step” button is replaced by a speed slider. For a similar effect, try moving the slider to the far left and watching the artist go through each step very slowly.

Example of level with “Step” button.

Example of level with speed slider..



Example level with no “Step” button.

Finger/Paper Tracing

For puzzles that don’t have a “Step” button, students can find a great amount of help looking through their program on their own.

The ability to predict what a program will do is an amazing skill, but it takes a lot of practice. Students can find opportunities to grow in this area by using finger tracing and/or paper tracing as they code in CS Fundamentals.

- Finger Tracing
    - Use your finger to point to the first block on the screen. What does everything look like? What values do the variables hold (if there are any)? What happens to that landscape after the current block is executed?
    - Drop your finger to the next block and ask the same questions.
    - Can you run through the whole program and predict the outcome?
- Paper Tracing
    - Draw a line down the center of your paper
    - On one side, write the code for your program
    - On the other side, write notes on the starting state of the program: Where are the characters? What are the values of the variables? Anything else to note?
    - Now look at the first line of your program. What changes when it is executed? Write the new details in your notes area.
    - Continue to run through the written code until you reach the end. Is everything as you expected?

Critical Questions

If looking through your code does not provide the answers you need, you will have to dig a little more deeply. These questions will help students to highlight the clues provided by their program when things are not working:

- What does it do?
- What is it supposed to do?
- What does that tell me?
- Is it right at the first step?
- Is it still right at the second step?
- Where is the first place that it goes wrong?

Asking for Help

In computer science, there's a fine line between the benefit that you get from figuring out something on your own, and the benefit of learning from the experience of a peer. For this reason, CS Fundamentals encourages a "Try three / Try three / Then ask me" system.

With this system (3/3/Me) students are asked to tackle a problem with three different solutions before asking peers for help. Once their classmates get involved, they should try three more solutions together before involving a teacher. At that point, as an educator, you should ask them to put in words what they tried and what that told them. Also ask them if they have narrowed down any places where the problem is **not**. If they have not yet solved their problem, guide them in trying a few additional options.

In all cases, make sure that you and your class are familiar with the right way to help a student:

- Don't sit in the classmate's chair
- Don't use the classmate's keyboard or tablet
- Don't touch the classmate's mouse
- Make sure the classmate can describe the solution to you out loud before you walk away

If, after all of this, issues still persist, you might consider filing a bug report with Code.org.

Rethinking Classroom Strategies

Ditch the Uniformity

Students learn at different rates. They also come into technology with vastly different skills. Trying to keep everyone on the same page will alienate both the bottom third and top third of learners. Take the pressure off of everyone by having a list of "approved" activities to focus on when they've finished their class exercise. These should include the Lesson Extras at the end of each CS Fundamentals lesson.

Frequent Breaks

Teachers are used to helping their class get very focused and encouraging students to work quietly until an activity is done. In computer science, students often benefit from small and frequent breaks, even if it's just switching to a new activity for a few minutes. Try having a student write a sentence or two about what they're trying to do or keep a notebook, like a biologist or chemist might.

Collaborate

It's really hard for a programmer to "cheat". Collaboration is a requirement out in the real world. This means helping one another solve problems, researching issues on the internet, and looking at what others have done in similar situations. The only bad method is claiming another's work as your own.

Don't be a Know-It-All

We often think that being a teacher means being an expert. In computer science, it's really much more important to be a cheerleader. Let the students know that it's possible for them to quickly become better at this than you are. Foster determination. Encourage students to monitor themselves and find answers for one another. Let them figure things out for themselves, then let them teach you.

Standards Mapping

CS Fundamentals was written using both the K–12 Computer Science Framework [https://k12cs.org] and the draft CSTA standards as guidance. Because the revised CSTA standards have not been finalized as of June 2017, we are waiting to publish formal standards mapping documents for CS Fundamentals to ensure that we have an opportunity to address any changes that may appear in the final draft. Once the CSTA standards have been finalized and published, we will complete a full pass to articulate, on a course and lesson level, our mapping to not only the the CSTA standards, but also to select ISTE, Common Core Math, Common Core ELA, and NGSS standards. Once this mapping has been completed, it will be available in the lesson plans and at code.org/csf/standards.

Assessments

When educating students, we find that assessments are often needed for a couple of different reasons: to check for comprehension and to determine grades. At Code.org, we believe that you know your students best. That's why we do not attempt to make any determination of what they should receive as a letter grade. Instead, we provide teachers with assistance in collecting examples of student comprehension. The ability to see where a student is succeeding and where they need help is fundamental to providing the opportunity to tailor their learning experience. That's why there's an ever-evolving teacher dashboard to highlight the work done by your class sections. Keep an eye on the Code.org blog [http://blog.code.org] for more information on changes and improvements.

Please note, we have provided assessment worksheets with each unplugged activity, and "assessment" puzzles for many online lessons. For more information on assessing student work using those items, see the thread on our teacher forum at http://forum.code.org/t/assessment-in-csf.

Outline of Courses A-F

Course A Overview (Ages 5+)

Course A offers computer science curriculum for beginning readers around the kindergarten age range. Students will learn to program using commands like loops and events. The lessons featured in this course also teach students to collaborate with others meaningfully, investigate different problem-solving techniques, persist in the face of difficult tasks, and learn about internet safety. By the end of this course, students create their very own custom game or story from Play Lab that they can share.

Course A Lesson Sequence

Online lessons are in regular text and unplugged lessons are in **bolded** text.

#	Lesson Name	Description
1	**Debugging: Unspotted Bugs**	This lesson will guide students through the steps of debugging. Students will learn the mantra: "What happened? What was supposed to happen? What does that tell you?"
2	**Persistence & Frustration: Stevie and the Big Project**	When students run into a barrier while answering a question or working on a project, it's so easy for them to get frustrated and give up. This lesson will introduce students to the idea that frustration can be an important part of learning. Here, frustration is presented as a step in the creative process, rather than a sign of failure.
3	**Real-Life Algorithms: Plant a Seed**	In this lesson, students will relate the concept of algorithms back to everyday, real-life activities by planting an actual seed. The goal here is to start building the skills to translate real-world situations to online scenarios and vice versa.
4	Learn to Drag and Drop	This lesson will give students an idea of what to expect when they head to the computer lab. It begins with a brief discussion introducing them to computer lab manners, then they will progress into using a computer to complete online puzzles.
5	**Programming Unplugged: Happy Maps**	The bridge from algorithms to programming can be a short one if students understand the difference between planning out a sequence and encoding that sequence into the appropriate language. This activity will help students gain experience reading and writing in shorthand code.

6	Programming in Maze	In this series of online puzzles, students will build on the understanding of algorithms, debugging, and general computer literacy. Featuring characters from the game Angry Birds, students will develop sequential algorithms to get the bird to the pig without crashing into walls or TNT. Debugging puzzles have also been mixed into this stage for added practice with problem solving and critical thinking.
7	**Common Sense Education: Going Places Safely**	In collaboration with Common Sense Education, this lesson helps students learn that many websites ask for information that is private and discusses how to responsibly handle such requests. Students also find out that they can go to exciting places online, but they need to follow certain rules to remain safe.
8	**Loops Unplugged: Happy Loops**	Loops can be a very helpful and powerful tool in programming. To understand how helpful loops are, students will be driven to want an easier way to solve mundane problems.
9	Loops in Collector	Building on the concept of repeating instructions from ‘Happy Loops’, this stage will have students using loops to collect treasure more efficiently on Code.org.
10	Loops in Artist	Returning to loops, students learn to draw images by looping simple sequences of instructions. In the previous plugged lesson, loops were used to traverse a maze and collect treasure. Here, loops are creating patterns. At the end of this stage, students will be given the opportunity to create their own images using loops.
11	**Events Unplugged: The Big Event**	Events are a great way to add variety to a pre-written algorithm. Sometimes you want your program to be able to respond to the user exactly when the user wants it to. That is what events are for.
12	Events in Play Lab	In this online activity, students will have the opportunity to learn how to use events in Play Lab and to apply all the coding skills they've learned to create an animated game. It's time to get creative and make a story in the Play Lab!

Course B Overview (Ages 6+)

Course B was developed with first graders in mind. Tailored to a novice reading level, this course also assumes limited knowledge of shapes and numbers.

At the moment, Course B closely parallels Course A, but provides more complex unplugged activities and more variety in puzzles. Students will learn the basics of programming, collaboration techniques, investigation and critical thinking skills, persistence in the face of difficulty, and internet safety. At the end of this course students will create their very own custom game from Play Lab that they can share with a link.

Course B Lesson Sequence

Online lessons are in regular text and unplugged lessons are in **bolded** text.

#	Lesson Name	Description
1	**Debugging: Unspotted Bugs**	This lesson will guide students through the steps of debugging. Students will learn the mantra: "What happened? What was supposed to happen? What does that tell you?"
2	**Persistence & Frustration: Stevie and the Big Project**	When students run into a barrier while answering a question or working on a project, it's so easy for them to get frustrated and give up. This lesson will introduce students to the idea that frustration can be an important part of learning. Here, frustration is presented as a step in the creative process, rather than a sign of failure.
3	**Real-Life Algorithms: Plant a Seed**	In this lesson, students will relate the concept of algorithms back to everyday, real-life activities by planting an actual seed. The goal here is to start building the skills to translate real-world situations to online scenarios and vice versa.
4	Learn to Drag and Drop	Learning to drag and drop online gives students an idea of what to expect when they head to the computer lab. It begins with a brief discussion introducing them to computer lab manners, then they will progress into using a computer to complete online puzzles.
5	**Common Sense Education: Your Digital Footprint**	In collaboration with Common Sense Education, this lesson helps students learn about the similarities of staying safe in the real world and when visiting websites. Students will also learn that the information they put online leaves a digital footprint or "trail." This trail can be big or small, helpful or hurtful, depending on how they manage it.

6	**Programming Unplugged: My Robotic Friends**	Using a predefined symbol key, your students will figure out how to guide one another to accomplish specific tasks without using any verbal commands. This segment teaches students the connection between symbols and actions, the difference between an algorithm and a program, and the valuable skill of debugging.
7	Programming in Maze	In this series of online puzzles, students will build on the understanding of algorithms, debugging, and general computer literacy. Featuring characters from the game Angry Birds, students will develop sequential algorithms to get the bird to the pig without crashing into walls or TNT. Debugging puzzles have also been mixed into this stage for added practice with problem solving and critical thinking.
8	More Programming in Maze	In this lesson, students will use their newfound programming skills in more complicated ways to navigate a tricky course.
9	**Loops Unplugged: My Loopy Robotic Friends**	Here, students learn the simplicity and utility of loops by “programming” their friends using the language from ‘My Robotic Friends’. Once loops are introduced, students find that they can build big structures faster.
10	Loops in Collector	Building on the concept of repeating instructions from ‘My Loopy Robotic Friends’, this stage will have students using loops to collect treasure more efficiently on Code.org.
11	Loops in Artist	Returning to loops, students learn to draw images by looping simple sequences of instructions. In the previous online lesson, loops were used to traverse a maze and collect treasure. Here, students use loops to create patterns. At the end of this stage, students will be given the opportunity to create their own images using loops.
12	**Events Unplugged: The Big Event**	Events are a great way to add variety to a pre-written algorithm. Sometimes you want your program to be able to respond to the user exactly when the user wants it to. That is what events are for.
13	Events in Play Lab	In this online activity, students will have the opportunity to learn how to use events in Play Lab and apply all of the coding skills that they've learned to create an animated game. It's time to get creative and make a story in Play Lab!

Course C Overview (Ages 7+)

Course C was developed for students in and around the second grade. It uses a limited understanding of shapes and elementary math concepts.

Students will create programs with loops, events, and conditionals. They will translate their initials into binary, investigate different problem-solving techniques, and discuss how to respond to cyberbullying. By the end of the course, students will create interactive games that they can share. Each concept in Course C is taught from the beginning, graduating toward experiences that allow for growth and creativity to provide all students a rich and novel programming experience.

Course C Lesson Sequence

Online lessons are in regular text and unplugged lessons are in **bolded** text.

#	Lesson Name	Description
1	**Building a Foundation**	In this lesson, students will build a structure with common materials. The structure will be tested on its ability to hold a textbook for more than ten seconds. Most students will not get this right the first time, but it's important they push through and keep trying.
2	Programming in Maze	Featuring characters from the game Angry Birds, students will develop sequential algorithms to move a bird from one side of the maze to a pig at the other side. To do this they will stack code blocks together in a linear sequence to move straight and turn left or right.
3	Debugging in Maze	Debugging is an essential element of learning to program. In this lesson, students will encounter puzzles that have been solved incorrectly. They will need to step through the existing code to identify errors, including incorrect loops, missing blocks, extra blocks, and blocks that are out of order.
4	**Real-Life Algorithms: Paper Airplanes**	In this lesson, students will relate the concept of algorithms back to everyday activities. After discussing algorithms, students will make paper airplanes using an algorithm built by a classmate. The goal here is to start creating the skills to translate real world situations to online scenarios and vice versa.
5	Programming in Collector	Students will continue to develop their understanding of algorithms and debugging in this series of puzzles by creating sequential algorithms to pick up treasure with a new character, Laurel the adventurer.

6	Programming in Artist	In this lesson, students will take control of the Artist to complete drawings on the screen. This Artist stage will allow students to create images of increasing complexity using new blocks like `“move forward by 100 pixels`” and `“turn right by 90 degrees.`”
7	**Getting Loopy**	Loops are a handy way to repeat actions a certain number of times. In this lesson, students will dance their way to a better understanding of how to use repeat loops.
8	Loops in Maze	Building on the concept of repeating instructions from “'Getting Loopy”', this stage will have students using loops to traverse mazes more efficiently than before.
9	Loops in Artist	Watch student faces light up as they make their own gorgeous designs using a small number of blocks and digital stickers! This lesson builds on the understanding of loops from previous lessons and gives students a chance to be truly creative. This activity is fantastic for producing artifacts for portfolios or parent/teacher conferences.
10	Loops in Harvester	In the preceding stage, students used loops to create fantastic drawings. Now they're going to loop new actions in order to help the harvester collect multiple veggies growing in large bunches.
11	**Events Unplugged: The Big Event**	Students will soon learn that events are a great way to add flexibility to a pre-written algorithm. Sometimes you want your program to be able to respond to the user exactly when the user wants it to. Events can make your program more interesting and interactive.
12	Build a Flappy Game	In this special stage, students get to build their own Flappy Bird game by using event handlers to detect mouse clicks and object collisions. At the end of the level, students will be able to customize their game by changing the visuals or rules.
13	Events in Play Lab	In this online activity, students will have the opportunity to learn how to use events in Play Lab and to apply all the coding skills they've learned to create an animated game. It's time to get creative and make a game in Play Lab!

14	**Common Sense Education: Screen Out the Mean**	This lesson helps children to recognize that it is essential to tell a trusted adult if something online makes them feel angry, sad, or scared. Students learn that other people can sometimes act like bullies when they are online. They will explore what cyberbullying means and what they can do when they encounter it. After reading a scenario about mean online behavior, students discuss what cyberbullying is, how it can make people feel, and how to respond. Finally, they use their knowledge to create a simple tip sheet on cyberbullying in their journals.
15	**Binary Bracelets**	Binary is extremely important in the world of computers. The majority of computers today store all sorts of information in binary form. This lesson helps demonstrate how it is possible to take something from real life and translate it into a series of ons and offs.

Course D Overview (Ages 8+)

Course D was created for students who read at roughly a third grade level. Angles and mathematical concepts are introduced with helpful videos and hints.

The course begins with a review of the concepts found in Courses A, B, and C. This review helps introduce or refresh basic ideas such as repeat loops and events. Students will develop their understanding of algorithms, nested loops, while loops, conditionals, and events. Lessons on digital citizenship are also included. This course is crafted to build a strong foundation of basic concepts before opening up to a wide range of new and exciting topics.

Course D Lesson Sequence

Online lessons are in regular text and unplugged lessons are in **bolded** text.

#	Lesson Name	Description
1	**Algorithms Unplugged: Graph Paper Programming**	By "programming" one another to draw pictures, students will develop an understanding of the connection between code and what it produces. The class will begin by having students instruct each other to color squares on graph paper in an effort to reproduce an existing picture. If there's time, the lesson can conclude with images that the students create themselves.
2	Introduction: Remembering Ideas from Course C	In this set of puzzles, students will begin with an introduction (or review depending on the experience of your class) of Code.org's online workspace. There will be videos pointing out the basic functionality of the workspace including the "Run", "Reset", and "Step" buttons. Also discussed in these videos: dragging, deleting, and connecting Blockly blocks. Next, students will practice their sequencing and debugging skills in maze. From there, students will see new types of puzzles like Collector, Artist, and Harvester when they learn the basics of loops.
3	Events in Bounce	In this online activity, students will learn what events are and how computers use them in programs. Students will work through puzzles, making their programs react to events (like arrow buttons being pressed.) At the end of the puzzle, students will have the opportunity to customize their game with different speeds and sounds.

4	Nested Loops in Maze	In this online activity, students will have the opportunity to push their understanding of loops to a whole new level. Playing with Bee and Plants vs Zombies, students will learn how to nest loops. They will also be encouraged to figure out how little changes will affect their program when they click "Run".
5	Nested Loops in Artist	Students will create intricate designs using Artist in this set of puzzles. By continuing to practice nested loops with new goals, students will see more uses of loops in general. This set of puzzles also offers more potential for creativity, with an opportunity for students to create their own design at the end of the stage.
6	Nested Loop Projects in Artist	Now that students know how to layer their loops, they can create so many beautiful things. This lesson will take students through a series of exercises to help them create their own portfolio-ready images.
7	**Debugging Unplugged: Relay Programming**	This activity will begin with a short review of 'Graph Paper Programming', then will quickly move to a race against the clock as students break into teams and work together to write a program one instruction at a time.
8	Debugging in Collector	In this online lesson, students will practice debugging in the Collector environment. Students will get to practice reading and editing code to fix puzzles with simple algorithms, loops and nested loops.
9	While Loops in Farmer	By the time students reach this lesson, they should already have plenty of practice using repeat loops, so now it's time to mix things up by learning a new structure, the `while` loop.
10	**Conditionals with Cards**	This lesson demonstrates how conditionals can be used to tailor a program to specific information. We don’t always have all the information we need when writing a program. Sometimes you will want to do something different in one situation than in another, even if you don't know what situation will be true when your code runs. That is where conditionals come in. Conditionals allow a computer to make a decision, based on the information that is true each time your code is run.
11	Conditionals in Bee	Up until this point students have been writing code online that executes exactly the same way each time it is run - reliable, but not very flexible. In this lesson, your class will begin to code with conditionals, allowing them to write programs that functions differently depending on the specific conditions the program encounters.

12	Conditionals & Loops in Maze	In this stage, students will be pairing together two key concepts: loops and conditionals. This set of puzzles bridges the gaps in understanding that occur when working on puzzles that use multiple kinds of blocks. By bringing two ideas together, students will create more complex code that shows both impressive creativity and critical thinking!
13	Conditionals & Loops in Farmer	Students will practice while loops, until loops, and if / else statements. All of these blocks use conditions. By practicing all three, students will learn to write complex and flexible code.
14	**Common Sense Education: Digital Citizenship**	In collaboration with Common Sense Education, this lesson helps students learn to think critically about the user information that some websites request or require. Students learn the difference between private information and personal information, distinguishing what is safe and unsafe to share online.
15	Build a Play Lab Game	In this online activity, students will have the opportunity to learn how to use events in Play Lab and to apply all of the coding skills they've learned to create an animated game. It's time to get creative and make a game in Play Lab!
16	**Binary Images**	Though many people think of binary as strictly zeros and ones, students will be introduced to the idea that information can be represented in a variety of binary options. This lesson takes that concept one step further as it illustrates how a computer can store even more complex information (such as photos and colors) in binary, as well.
17	Binary in Artist	Binary is extremely important in the world of computers. The majority of computers today store all sorts of information in binary form. In this lesson, students will build binary images in Artist by translating 0s and 1s to offs and ons (or blacks and whites).

Course E Overview (Ages 9+)

Created with fourth grade students in mind, this course begins with a brief review of concepts previously taught in courses C and D. This introduction is intended to inspire beginners and remind the experts of the wonders of computer science. Students will practice coding with algorithms, loops, conditionals, and events before they are introduced to functions. At the end of the course, students will have the opportunity to create a capstone project that they can proudly share with peers and loved ones.

Course E Lesson Sequence

Online lessons are in regular text and unplugged lessons are in **bolded** text.

#	Lesson Name	Description
1	**Dice Race**	In this lesson, students will relate the concept of algorithms back to real-life activities by playing the Dice Race game. The goal here is to start building the skills to translate real-world situations to online scenarios and vice versa.
2	Introduction (Course Warm Up)	In this progression, students will begin with an introduction (or review depending on the experience of your class) of Code.org's online workspace. Students will learn the basic functionality of the interface including the “Run”, “Reset”, and “Step” buttons. Dragging, deleting, and connecting Blockly blocks is also introduced in the beginning video. In the puzzles, students will practice their sequencing and debugging skills in Maze and Artist.
3	Intro to Conditionals	This lesson introduces students to while loops and if/else statements. While loops are loops that continue to repeat commands as long as a condition is true. While loops are used when the programmer doesn't know the exact number of times the commands need to be repeated, but the programmer does know what condition needs to be true in order for the loop to continue looping. If/Else statements offer flexibility in programming by running entire sections of code only if something is true, otherwise it runs something else.

4	**Common Sense Education: Personal and Private Information**	Developed by Common Sense Education, this lesson is about the difference between information that is safe to share online and information that is not. As students visit sites that request information about their identities, they learn to adopt a critical inquiry process that empowers them to protect themselves and their families from identity theft. In this lesson, students learn to think critically about the user information that some websites request or require. They learn the difference between private information and personal information, as well as how to distinguish what is safe or unsafe to share online.
5	Build a Star Wars Game	In this lesson, students will practice using events to build a game that they can share online. Featuring R2-D2 and other Star Wars characters, students will be guided through events, then given space to create their own game.
6	**Functions Unplugged: Songwriting**	One of the most magnificent structures in the computer science world is the function. Functions (sometimes called procedures) are mini programs that you can use over and over inside of your bigger program. This lesson will help students intuitively understand why combining chunks of code into functions can be such a helpful practice.
7	Functions in Artist	Students will be introduced to using functions on Code.org. Magnificent images will be created and modified with functions in Artist. For even more complicated patterns, students will learn about nesting functions by calling one function from inside of another.
8	Functions in Bee	In the second round of practice with online functions, students will navigate complex paths, collect plenty of nectar, and make tons of honey.
9	Functions in Farmer	Students have practiced creating impressive designs in Artist and navigating mazes in Bee, but today they will use functions to harvest crops in Farmer. This lesson will push students to use functions in new ways by combining them with while loops and if/else statements.
10	Determine the Concept	This lesson brings together concepts from the previous lessons and gives students a chance to think critically about how they would solve each problem, but without telling them which concept to apply. Students will review basic algorithms, debugging, repeat loops, conditionals, while loops, and functions.

11	Build a Play Lab Game	This lesson features Play Lab, a platform where students can create their own games and have interactions between characters and user input. Students will work with events to create keyboard controls. This set of puzzles will also loosely guide students through game development, but with freedom to add their own ideas.
12	Explore Project Ideas	The next four lessons provide an opportunity for students to put their coding skills to use in a capstone project. This project will help individuals gain experience with coding and produce an exemplar to share with peers and loved ones. Intended to be a multi-lesson or multi-week experience, students will spend time exploring, brainstorming, learning about the design process, building, and presenting their final work. This first week, students will play with pre-built examples of projects in both Artist and Play Lab for inspiration.
13	The Design Process	In this portion of the project, students will learn about the design process and how to implement it in their own projects.
14	Build Your Project	Now the students will be given their own space to create their project with either Artist or Play Lab. This will be the longest portion of the project.
15	Present Your Project	Finally, students will be able to present their finished work to their peers or share with their loved ones with a special link.
16	**The Internet**	Even though many people use the internet daily, not very many know how it works. In this lesson, students will pretend to flow through the internet, all the while learning about connections, URLs, IP Addresses, and the DNS.
17	**Crowdsourcing**	In computer science, we face some big, daunting problems. Challenges such as finding large prime numbers or sequencing DNA are almost impossible to do as an individual. Adding the power of others makes these tasks manageable. This lesson will show your students how helpful teamwork can be in the industry of computer science.

Course F Overview (Ages 10+)

The last course in CS Fundamentals was tailored to the needs students in the fifth grade.

In these lessons, students will create programs with different kinds of loops, events, functions, and conditionals. They will also investigate different problem-solving techniques and discuss societal impacts of computing and the internet. By the end of the curriculum, students create interactive stories and games that they can share with their friends and family.

Course F Lesson Sequence

Online lessons are in regular text and unplugged lessons are in **bolded** text.

#	Lesson Name	Description
1	**Algorithms Unplugged: Tangrams**	This lesson shows us something important about algorithms. As long as you keep an algorithm simple, there are lots of ways to use it. However, if you want to make sure everyone produces the same outcome, then your algorithm needs more detail. Students will learn the difference between a detailed and general algorithm while playing with tangrams.
2	Introduction (Course Warm Up)	In this lesson, students will be introduced to sequences, loops, and nested loops to prepare them for more complicated concepts in the later part of the course. This "ramp up" lesson equalizes the playing field between the experts and the beginners in your class.
3	**Common Sense Education: The Power of Words**	Students consider that while they are enjoying their favorite websites they may encounter messages from other kids that can make them feel angry, hurt, sad, or fearful. They explore ways to handle cyberbullying and how to respond in the face of upsetting language online.
4	Events in Ice Age	In this lesson, students are guided through a story featuring characters from Ice Age. Students will work with events and loops to make characters move on the screen, and will get the chance to create their own game or story after the guided levels.
5	Conditionals in Minecraft	This lesson gives students a chance to learn and practice conditionals. It features characters and settings from Minecraft, and students will complete tasks such as mining and building structures using their programs.

6	**Variables Unplugged: Envelope Variables**	Variables are used as placeholders for values such as numbers or words. Variables allow for a lot of freedom in programming. Instead of having to type out a phrase many times or remember an obscure number, computer scientists can use variables to reference them. This lesson helps to explain what variables are and how we can use them in many different ways. The idea of variables isn't an easy concept to grasp, so we recommend allowing plenty of time for discussion at the end of the lesson.
7	Variables in Artist	In this lesson, students will explore the creation of repetitive designs using variables in the Artist environment. Students will learn how variables can be used to make code easier to write and easier to read. After guided puzzles, students will end in a freeplay level to show what they have learned and create their own designs.
8	Variables in Play Lab	Students will get further practice with variables in this lesson by creating scenes in Play Lab. Students will work with user input to set the values of their variables, then get space to create their own mini-project with variables.
9	**For Loops Unplugged: For Loop Fun**	We know that loops allow us to do things over and over again, but now we're going to learn how to use loops that have extra structures built right in. These new structures will allow students to create code that is more powerful and dynamic.
10	For Loops in Bee	Featuring Bee, this lesson focuses on for loops and using an incrementing variable to solve more complicated puzzles. Students will begin by reviewing loops from previous lessons, then walked through using for loops to more effectively solve complicated problems.
11	For Loops in Artist	In this lesson, students continue to practice for loops, this time with Artist. Students will complete puzzles combining the ideas of variables, loops, and for loops to create complex designs. At the end, they will have a chance to create their own art in a freeplay level.
12	**Functions Unplugged: Songwriting with Parameters**	One of the most magnificent structures in the computer science world is the function. Functions (sometimes called procedures) are mini programs that you can use over and over inside of your bigger program. This lesson will help students intuitively understand why combining chunks of code into functions is such a helpful practice, and how they can use those structures even when chunks of code are slightly different.

13	Functions in Bee	This lesson teaches students how to create simple functions using our sophisticated “modal” function editor, preparing the way for them to incorporate parameters in future lessons.
14	Functions with Parameters in Artist	In this lesson, students continue working with functions with and without parameters. Students will get the chance to create their own drawings with and without parameters, before modifying functions in a freeplay level.
15	Functions with Parameters in Bee	This lesson features the bee environment, and continues along the concept of functions with parameters from the previous Artist stage. Students will practice writing and using functions to follow complex paths and collect patterns of nectar and honey.
16	Explore Project Ideas	The next five lessons provide an opportunity for students to put their coding skills to use in a capstone project. This project will help individuals gain experience with coding and produce an exemplar to share with peers and loved ones. This is intended to be a multi-lesson or multi-week project where students spend time brainstorming, learning about the design process, building, and then presenting their final work. This first week, students will play with pre-built examples of projects in both Artist and Play Lab for inspiration.
17	The Design Process	In this portion of the project, students will learn about the design process and how to implement it in their own projects.
18	Build Your Project	Now the students will be given their own space to create their project with either Artist or Play Lab. This will be the longest portion of the project.
19	Revise Your Project	Now that the projects are built, students are given the opportunity to get feedback from peers and revise their projects.
20	Present Your Project	Finally, students will be able to present their finished work to their peers or share with their loved ones with a special link.

Courses A-F Additional Supplies List

Most of the unplugged lessons in Courses A-F require only paper and writing instruments. We have included here a list of courses where lessons

Course A

- ❏ **Marble Run** - Tape, cardboard, other misc. supplies for building, kid friendly marbles (or round cereal)
- ❏ **Real-Life Algorithms: Plant a Seed** - Dirt, seeds, paper cups
- ❏ Scissors for each student

Course B

- ❏ **Marble Run** - Tape, cardboard, other misc. supplies for building, kid friendly marbles (or round cereal)
- ❏ **Real-Life Algorithms: Plant a Seed** -Dirt, seeds, paper cups
- ❏ **My (Loopy) Robotic Friends** - Plastic cups
- ❏ Scissors for each student

Course C

- ❏ **Building a Foundation** - Gumdrops and toothpicks or marshmallows and popsicle sticks
- ❏ **Real-Life Algorithms: Paper Airplanes** - Paper for airplane construction
- ❏ **Binary Bracelets** - Markers (Optional: beads and pipe cleaners)
- ❏ Scissors for each student

Course D

- ❏ **Conditionals with Cards** - Deck of cards, or something similar
- ❏ **Digital Citizenship** - Cubecraft superheros (see resources for more details)
- ❏ **Binary Images** - Optional: Objects with two opposites such as a large coin or a lamp that turns on or off
- ❏ Scissors for each student

Course E

- ❏ **Dice Race** - Dice
- ❏ **Crowdsourcing** - Deck of cards, or something similar and a jar full of small items (buttons, beads, beans, etc)
- ❏ Scissors for each student

Course F

- ❏ **Envelope Variables** - Envelopes
- ❏ **For Loop Fun** - Dice
- ❏ Scissors for each student

Appendix A: Unplugged Lesson Plans

(With complete list of supplies beforehand)

Course A

Lesson 1: Debugging: Unspotted Bugs

Bug | Debugging | Persistence | Unplugged

Overview

This lesson will guide students through the steps of debugging. Students will learn the mantra: "What happened? What was supposed to happen? What does that tell you?"

Purpose

Research shows that some students have less trouble debugging a program than writing one when they first learn to code. In this lesson, we introduce the idea of debugging in a real world sense.

The goal in this lesson is to teach students steps to spot a bug and to increase persistence by showing them that it's normal to find mistakes. In later lessons, students will debug actual programs on Code.org.

Agenda

Warm Up (12 min)

- **Unspotted Bugs Vocabulary**

Marble Run Breakdown (10 - 20 min)

- **Debug the Run**

Wrap Up (10 - 20 min)

- **Journaling**

Extended Learning

- **Real Life Bug Hunting**

Objectives

Students will be able to:

- Express that they have noticed when something goes differently than what is expected.
- Identify what the expected result was before an error occurs.
- Determine and describe the difference between what was expected and what actually happened in the event of an error.

Preparation

- [ ] Review the Unspotted Bugs Story (**Unspotted Bugs - Online Story**)
- [ ] Pre-read Unspotted Bugs to identify appropriate questions for your classroom
- [ ] Follow instructions in the **Marble Run - Teacher Prep Guide** to make a Marble Run (which will be arranged incorrectly at the start)
- [ ] Give a **Think Spot Journal** to each student

Links

For the Teacher

- **Marble Run** - Teacher Prep Guide (**PDF** | **DOCX**)
- **Think Spot Journal** (**PDF** | **DOCX**)

For the Students

- **Unspotted Bugs** - Online Story

Vocabulary

- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in your algorithm or program.
- **Persistence** - Trying again and again, even when something is very hard.

Teaching Guide

Warm Up (12 min)

Goal: Help students understand the steps involved in debugging.

Unspotted Bugs

This story can be presented in several ways, including:

- Circled up story time
- Projected with document camera / smartboard
- Pair shared with students at their computers

The story of Unspotted Bugs presents many of the ideas that students will need to understand the debugging process of coding. This warm-up is meant to tie a memorable story together with a concept that young kids often find to be difficult.

Read the book and discuss the techniques that JD used to discover and take care of bugs. Make sure those questions and tactics get repeated often enough that students can recall (if not recite) them without the story in hand.

Lesson Tip

Important ideas from the story:

- What happened?
- What was supposed to happen?
- What does that tell you?
- Did it work at the first step?
- Did it work at the second step?
- Where did it go wrong?

Potential Questions for Storytime:

- Page 3: What do you notice in the picture? What's wrong with the flower? (It's upside down!) What's wrong with the clock? (The hands aren't in the center) Why do you think there is something wrong with these items?(Because there are bugs on them!)
- Page 7: What's wrong with the picture? (The lamp is upside down) Why is that? (There's a bug)
- Page 11: What's wrong in this scene? (The car doesn't have wheels!) Why? (Because there are bugs on it!)
- What did JD find when he went looking for the bug? What was wrong? What does this mean? (JD found an upside down tree. This is wrong because the tree trunk should be touching the ground! This means there is a bug on the tree!)

Vocabulary

This lesson has three new and important vocabulary words:

- *Bug* - Say it with me: Buhh-g. Something that is going wrong. An error.
- *Debugging* - Say it with me: Dee-bug-ing. To find and fix errors.
- *Persistence* - Say it with me: Purr-siss-tense. Not giving up. Persistence works best when you try things many different ways, many different times.

Marble Run Breakdown (10 - 20 min)

Goal: Help students think critically about the difference between what is happening and what is expected.

Debug the Run

Now that students have been introduced to the idea of looking for problems, they can try to apply it to more places in the real world. This next activity gives them practice looking for bugs in Marble Runs (a project that they will be working with next week.)

Grab your sample marble run (built from our plans, or something similar.) Show the students how each piece works, then demonstrate putting them together (but put them together incorrectly, to prevent the ball from flowing properly from A to B.

The goal of this exercise is to help the students identify when something goes wrong, so if they don't catch it the first time, run it again, and again. It can help to make exaggerated frustration faces when the ball doesn't do what you would like it to do.

Let the students share hypotheses about what is going wrong, and how to fix it. Students should feel free to try things that you know will be incorrect. If students misidentify solutions, use the bug finding formula on their configurations. Repeat until you get a working run.

Encouragement is key here. If things don't work right away, praise the class for being so persistent and choosing not to give up. If they start to get frustrated, encourage them to persist a bit longer, promising them that they will get it soon if they just hang in there.

Lesson Tip

Say:

Great! You all are so good at this, maybe you can help me with my own problem!

See, I have this marble run that I made. It comes in two pieces. When I put the ball in here (input A) it's supposed to come out here (output A). When I put the ball in here (input B) it's supposed to come out here (output B). Now, when I slide them together, I should be able to put the ball in here (input A) and have it come out here (output B). But it doesn't work, watch.

[Slide the pieces together with output B facing output A.]

Watch what happens. [Drop ball at input A and notice that it does not come out output B.]

- BUG!

What happened?

- The ball fell on the table.

What was supposed to happen?

- The ball was supposed to drop from A into B.

What does that tell you?

- You should turn B around so that the ball goes into the right place!

Wrap Up (10 - 20 min)

Journaling

Goal: Students will start to understand the importance of the activity they just completed by reflecting on it verbally, then through drawing in their journals.

Clear your mind:

It can be distracting to a learner when they have unanswered questions or doubts. To end this lesson, we'll give everyone the chance to get those out so that they can reflect on what they've been taught.

Encourage students to share their thoughts and questions either with the whole class or with an elbow partner.

Reflect:

Once they've had time to ponder their own thoughts, get the students thinking about the purpose of the lesson that they just learned. Why did you do this activity? How will it help them later? Can they think of buggy things that they've seen in the real world?

Students should finish by drawing or writing in their journal. Possible topics include:

Lesson Tip

Say:

What do you think we learned in this lesson?

- Debugging
- How to solve a problem
- How to make a marble go
- How do you think that can help us in other places?

- How do you feel when something that you are working on acts buggy?
- How many times do you think you should try to fix a bug before you give up?
- What would you do if you notice that something is buggy, but you don't know how to fix it?

Extended Learning

Real Life Bug Hunting

Take your students outside. Do you see any signs of bugs? What are they? Now look closer... can you find the actual bug?

Lesson Tip:

The signs of real-live bugs won't be as dramatic as upside down trees, but it might be dead leaves, spots on flowers, or slime on the
sidewalk. Have the students brainstorm these before going outside to look for them.

Marble Run

Teacher Prep Guide

This guide will provide assistance through a set of two lessons using a Marble Run contraption.

The first portion of this kindergarten series is the debugging lesson, where students will help you debug your Marble Run. In order to do this, you need to have a broken prototype that can be fixed in a predictable way. This guide will suggest an easy step-by-step solution, and give you tips for making a creation using your own design.

For the second half, we are going to ask students to do something incredibly challenging in order to stretch their understanding and aptitude for persistence. This guide will provide additional suggestions and resources to keep the project grade appropriate.

Stage 1: Debugging

The Rules:

The rules of the student version of the Marble Run activity are pretty simple:

1) Build two Marble Runs.
2) Each Marble Run should have at least 3 pieces.
3) Marble Run 1 should take a marble at **Start** height and finish at **Middle** height.
4) Marble Run 2 should take a marble at **Middle** height and finish at **End** height.
5) Put the two Marble Runs together and watch the marble go from **Start** to **End** .

There are a couple of additional rules to adapt this to be effective for the lesson on debugging:

1) The teacher's contraption must not work to begin with.
2) The fix for the issue should be detectable by following the marble's path and determining where the change from "expected" to "unexpected" occurs.

The Set-Up:

Use the Marble Run Ruler (provided on page 2) to determine the starting and ending height for each of the two components, we will call those Component A and Component B.

Component A needs to take in a marble (Input A) at a height that falls somewhere within the highlighted "Start" region. It should then return the marble (Output A) at a height somewhere within the highlighted "Middle" region.

Component B should take a marble (input B) at a height that falls somewhere within the highlighted "Middle" region. It should then return the marble (Output B) at a height somewhere within the highlighted "End" region.

Two simple ways for a teacher to initiate an easy-to-fix failure would be:

A) Have two working components, but connect them in an incorrect way
B) Have Component A release the marble lower than Component B can receive it

Proceed to the teacher guide for Stage 2 for more information on building a Marble Run that falls into either of those categories.

Start  Start

Middle  Middle

End  End

Stage 2: Building a Marble Run

The Rules:

These are the rules for the student version of the Marble Run activity:

1) Build two Marble Runs.
2) Each Marble Run should have at least 3 pieces.
3) Marble Run 1 should take a marble at **Start** height and finish at **Middle** height.
4) Marble Run 2 should take a marble at **Middle** height and finish at **End** height.
5) Put the two Marble Runs together and watch the marble go from **Start** to **End**.

Feel free to change the parameters of these heights as you see fit.

The Set-Up:

Set-up of the student resource area is crucial. Supplies should be plentiful and easy to locate. In addition to the classroom norms (cardstock, tape, safety scissors) volunteers can also donate extra items if given enough notice (paper cups, cereal boxes, and the like).

For further support, place a stack of copies of "Marble Run Hints" (pages 7 & 8) for students to find. You do not need to let the class know that those are available. Students will feel more like they have "discovered" something if the teacher is not involved in the process.

The Build:

We have provided tutorials on four relatively simple pieces that are quite helpful for this project. These peices are:

- **Tube (fig. 1)** - A piece of paper that has been rolled into a cylinder
- **Ramp (fig. 2)** - Paper folded in a zig-zag fashion to provide a ramp with attaching flaps
- **Bridge (fig. 3)** - Paper where two sides have been folded into the center to create a bridge
- **Cone (fig. 4)** - Paper rolled first into a cylinder, then tightened at the bottom and loosened at the top. Once the basic cone has been created, secure with tape, then cut the top and bottom to customize.

A low-frills example contraption can be created using the steps that follow.

Component A:

1) Cut an 8.5"x11" sheet of cardstock in half lengthwise, then cut one of those halves lengthwise again. Fold both of the quarter strips bridge style.
2) Lay the bridges on their sides and tape free edges together to form a square or rectangle.
3) Cut an 8.5"x11" sheet of cardstock in quarters (length, then width). Roll two of the pieces along the long edge and two along the short edge, then secure with tape, to make a total of 4 tubes.
4) Tape the long tubes to the back of the square case from step 2, and the short tubes should be taped in front.

Marble Run

Teacher Prep Guide

Component A (continued):

5) Cut an 8.5”x11” sheet of cardstock in half widthwise. Fold one piece in half lengthwise, then fold the long edges back out toward the crease to make a ramp.
6) Tape the edges of the ramp to the tops of the posts. This gives your main marble path, but it’s not quite tall enough.
7) Add a cone at the intake point, and you’ll be set!

Component B:

8) Cut an 8.5”x11” sheet of cardstock in half widthwise and roll one of the pieces along the short side, to make an 8.5” tube. Secure with tape.
9) Cut the tube at a point anywhere from 2” - 4” at about a 45 degree angle.
10) Rotate one of the pieces to form an elbow, and tape back together.
11) Cut a 1“ strip from the remaining 8.5” x 5.5” cardstock half (lengthwise) and make a bridge to use as a triangular base for the tube to sit in.
12) Use the remaining cardstock to make an input cone for the top of Component B. Trim tube and cone to get appropriate height.

Voila! Your very own Marble Run!

Note: It is highly unlikely that your students will come up with anything this clean and stable. That is OKAY! The version here is meant to be messed with and reused.

Tube

Figure 1

Ramp

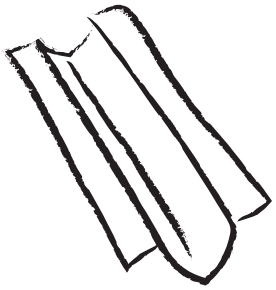


Figure 2

Bridge

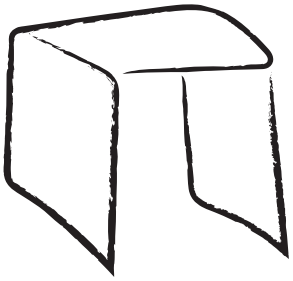


Figure 3

Cone

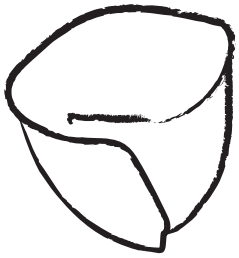


Figure 4

Lesson 2: Persistence & Frustration: Stevie and the Big Project

Fail | Frustrated | Persistence | Unplugged

Overview

When students run into a barrier while answering a question or working on a project, it's so easy for them to get frustrated and give up. This lesson will introduce students to the idea that frustration can be an important part of learning. Here, frustration is presented as a step in the creative process, rather than a sign of failure.

This lession can be done over one or two class sessions. If you have more time, feel free to draw out the building and revising phase of the Marble Run activity.

Purpose

The goal of this lesson is to help students realize that failure and frustration are common when working on projects, but that doesn't mean that they should give up.

In this lesson, students will develop an understanding of what it means to be frustrated while working on a large project. It's possible that not every student will experience frustration with this activity, but there are many opportunities to open a discussion about moments in the past where students have felt frustrated but nevertheless persisted.

Agenda

Warm Up (15 min)

- **Stevie and the Big Project**
- **Vocabulary**

Marble Run (20 - 45 min)

- **Before the Project**
- **Building the Marble Run**
- **After the Marble Run**

Wrap Up (5 min)

- **Journaling**

Extended Learning

Objectives

Students will be able to:

- Recognize and point out symptoms of frustration.
- Describe at least one reason why they will choose to be persistent in the face of frustration, rather than giving up.

Preparation

- [ ] Pre-read "Stevie and the Big Project" to identify appropriate questions for your class.
- [ ] Follow instructions in the **Marble Run - Teacher Prep Guide** to make a Marble Run.
- [ ] Print copies of the **Marble Run Ruler** (page 2 of teacher guide) for each student or pair of students.
- [ ] Prepare a resource station with cardstock, safety scissors, tape, and anything else you think might be fun for students to build with. Include a stack of the **"Marble Run Hints"** pages from the Teacher Prep Guide, but do not advertise their existence.
- [ ] (Optional) Allow students to bring cardboard, popsicle sticks, string, or other tidbits from home to add to the resource station.
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **Marble Run** - Teacher Prep Guide (**PDF** | **DOCX**)
- **Think Spot Journal** (**PDF** | **DOCX**)

For the Students

- **Stevie and the Big Project** - Online Story

Vocabulary

- **F.A.I.L.** - First Attempt In Learning
- **Frustrated** - Feeling annoyed or angry because something is not the way you want it.
- **Persistence** - Trying again and again, even when something is very hard.

Teaching Guide

Warm Up (15 min)

Stevie and the Big Project

Goal: Introduce students to the idea that they don't have to give up just because they are frustrated.

This lesson begins with a story. Students will be introduced to several ideas on persistence and frustration through relatable struggles by fictional characters, including the idea that frustration is not a sign that someone should instantly give up.

This book can be presented in several ways, including:

- Circled up story time
- Projected with document camera / smartboard
- Pair share with students at their computers

Use the reading techniques that work in your classroom:

If your students like to discuss things that happen as they appear in the book, be sure to stop your class after large plot areas like when Stevie breaks her structure, or when Laurel explains frustration.

If your students like to sit through a whole story and discuss at the end, read through the book, then prompt their memory with some "Remember when…" type questions.

Lesson Tip

Sample Questions:

- How would you feel if you were given a project that feels much harder than what you are used to?
- Do you think it's okay to try something new, even if it doesn't work out the first time?
- Why do you think Stevie smashed her project?
  - Do you think that helped her or hurt her when it comes to reaching her goal?
  - What do you think Stevie should have done instead of breaking her project?
- Can somebody explain what frustration is?
- How do you think you can know when you are frustrated?
  - What face do you make when you are frustrated?
  - How can you make yourself feel better when you start to get frustrated?
  - We all get frustrated sometimes. Does that mean that we should give up?
- Can someone tell me what persistence is?
  - Why is it hard to learn if you're not persistent?
  - Can you tell me why you might be tempted not to be persistent?
  - What happened when Stevie decided to be persistent?
  - Do you think you can be persistent?

Vocabulary

- *Persistence* - Say it with me: Purr-siss-tense. Not giving up. Persistence works best when you try things many different ways, many different times.
- *Frustrated* - Say it with me: Frus - straight - ted. Feeling annoyed or angry because something is not the way you want it.
- *F.A.I.L.* - First Attempt in learning. When you try to do something, but you don't do it quite right.

Marble Run (20 - 45 min)

Goal: This activity is meant to highlight and normalize the feeling of frustration, while giving students a chance to be persistent.

Before the Project

It is vitally important that students understand that this activity is meant to help them learn about frustration and persistence. This is not one of those times when we allow students to experience something, then give it a name afterward. Students need to know that they will be feeling some emotions, and that those emotions are okay.

Take a moment to relate the next activity back to the book that you just read. The class might be excited that they get to try the same project that Stevie did, but they might also be apprehensive at the thought of tackling something difficult.

Encourage your students to have their Think Spot Journals around during the activity so they can use them to plan, solve, and voice concerns.

Building the Marble Run

Time to be an engineer!

Break students up into pairs and have them quickly come up with a team name. This should help to unify them in their work.

Next, point out the resource station that you have set up with all of the supplies and goodies that students will have access to. Make sure you are very clear about whether they are limited only to the items in the resource station or whether they are allowed to ask for other items for their creation.

Give students checkpoints for this activity. Make sure that they know that there is no penalty for not finishing on time.

Preplanning is optional, since prediction is not often a kindergartener's strong suit.

The first attempt at building will likely be hectic and a bit sloppy, but it should give students access to the feelings and opportunities for persistence that are being studied in this lesson.

Try to end the Marble Run build with an opportunity for groups to collaborate. This will improve the chances of success for students who have been struggling, without the need for teacher intervention.

After the Marble Run

Time to do some damage control if any is needed.

Remind students that this activity was planned to teach students how to identify feelings of frustration and work past them to be persistent.

Discuss the difference between being successful for the purpose of this activity, and being successful at building their contraption. Is it possible to have done the first without the second?

Lesson Tip

Say:

Now, we're going to do something very fun, and very challenging! I am going to let you all try to make a Marble Run of your own!

This is **supposed** to be challenging. That's part of the fun! Your Marble Run probably won't work right the first time, and that's
alright. The goal for this game is to practice being persistent.

Remember, Stevie showed us that this might be difficult, and sometimes difficult things are frustrating. It is okay if you get frustrated during this activity. Most of us probably will at some point. How should we handle those feelings?

- Count to 10
- Take deep breaths
- Journal about them
- Talk to a partner about them
- Ask for help

Lesson Tip

Checkpoint Suggestions:

- Pre-planning time (3-5 minutes)
- First attempt at building (10-15 minutes) -- For a longer (or two day) time period --
- Discuss with another group (3-5 minutes)
- Revision of structure (10-15 minutes) -- Wrap Up Work --
- Collaborative work time (5-15 minutes)

Teacher Tip

Tears are a very common byproduct when kindergarteners attempt lessons of this nature. You will likely want to have a pre-packaged prescription for students who become emotionally raw.

- Can you put into words what you are feeling right now?
- Stevie would be so proud of you. What do you think Laurel and Jorge would say if you told them how you feel?
- What would it be called if you said out loud that you are frustrated, but decided to keep working anyway?
  - Do you feel like you can be persistent with me today?

Wrap Up (5 min)

Journaling

Goal: Allow students to reflect on the emotions and processes experienced during the lesson.

Journal Prompts:

Finish out this lesson by asking students to spend some time in their Think Spot Journal.

- Draw a picture of what you look like when you're frustrated.
- Draw a picture that shows things you can do to feel better when you're frustrated.
- What does persistence look like?

Extended Learning

- Add a third piece to the beginning of the Marble Run. Can students start a marble up even higher and get it to flow through the rest of their contraption?

- Talking through frustration. Can students think of things that they can say to classmates to help them be persistent when they are frustrated?

Marble Run

Teacher Prep Guide

This guide will provide assistance through a set of two lessons using a Marble Run contraption.

The first portion of this kindergarten series is the debugging lesson, where students will help you debug your Marble Run. In order to do this, you need to have a broken prototype that can be fixed in a predictable way. This guide will suggest an easy step-by-step solution, and give you tips for making a creation using your own design.

For the second half, we are going to ask students to do something incredibly challenging in order to stretch their understanding and aptitude for persistence. This guide will provide additional suggestions and resources to keep the project grade appropriate.

Stage 1: Debugging

The Rules:

The rules of the student version of the Marble Run activity are pretty simple:

1) Build two Marble Runs.
2) Each Marble Run should have at least 3 pieces.
3) Marble Run 1 should take a marble at **Start** height and finish at **Middle** height.
4) Marble Run 2 should take a marble at **Middle** height and finish at **End** height.
5) Put the two Marble Runs together and watch the marble go from **Start** to **End** .

There are a couple of additional rules to adapt this to be effective for the lesson on debugging:

1) The teacher's contraption must not work to begin with.
2) The fix for the issue should be detectable by following the marble's path and determining where the change from "expected" to "unexpected" occurs.

The Set-Up:

Use the Marble Run Ruler (provided on page 2) to determine the starting and ending height for each of the two components, we will call those Component A and Component B.

Component A needs to take in a marble (Input A) at a height that falls somewhere within the highlighted "Start" region. It should then return the marble (Output A) at a height somewhere within the highlighted "Middle" region.

Component B should take a marble (input B) at a height that falls somewhere within the highlighted "Middle" region. It should then return the marble (Output B) at a height somewhere within the highlighted "End" region.

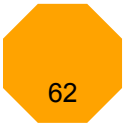
Two simple ways for a teacher to initiate an easy-to-fix failure would be:

A) Have two working components, but connect them in an incorrect way
B) Have Component A release the marble lower than Component B can receive it

Proceed to the teacher guide for Stage 2 for more information on building a Marble Run that falls into either of those categories.

Start  Start

Middle  Middle

End  End

Stage 2: Building a Marble Run

The Rules:

These are the rules for the student version of the Marble Run activity:

1) Build two Marble Runs.
2) Each Marble Run should have at least 3 pieces.
3) Marble Run 1 should take a marble at **Start** height and finish at **Middle** height.
4) Marble Run 2 should take a marble at **Middle** height and finish at **End** height.
5) Put the two Marble Runs together and watch the marble go from **Start** to **End**.

Feel free to change the parameters of these heights as you see fit.

The Set-Up:

Set-up of the student resource area is crucial. Supplies should be plentiful and easy to locate. In addition to the classroom norms (cardstock, tape, safety scissors) volunteers can also donate extra items if given enough notice (paper cups, cereal boxes, and the like).

For further support, place a stack of copies of "Marble Run Hints" (pages 7 & 8) for students to find. You do not need to let the class know that those are available. Students will feel more like they have "discovered" something if the teacher is not involved in the process.

The Build:

We have provided tutorials on four relatively simple pieces that are quite helpful for this project. These peices are:

- **Tube (fig. 1)** - A piece of paper that has been rolled into a cylinder
- **Ramp (fig. 2)** - Paper folded in a zig-zag fashion to provide a ramp with attaching flaps
- **Bridge (fig. 3)** - Paper where two sides have been folded into the center to create a bridge
- **Cone (fig. 4)** - Paper rolled first into a cylinder, then tightened at the bottom and loosened at the top. Once the basic cone has been created, secure with tape, then cut the top and bottom to customize.

A low-frills example contraption can be created using the steps that follow.

Component A:

1) Cut an 8.5"x11" sheet of cardstock in half lengthwise, then cut one of those halves lengthwise again. Fold both of the quarter strips bridge style.
2) Lay the bridges on their sides and tape free edges together to form a square or rectangle.
3) Cut an 8.5"x11" sheet of cardstock in quarters (length, then width). Roll two of the pieces along the long edge and two along the short edge, then secure with tape, to make a total of 4 tubes.
4) Tape the long tubes to the back of the square case from step 2, and the short tubes should be taped in front.

Marble Run

Teacher Prep Guide

Component A (continued):

5) Cut an 8.5”x11” sheet of cardstock in half widthwise. Fold one piece in half lengthwise, then fold the long edges back out toward the crease to make a ramp.
6) Tape the edges of the ramp to the tops of the posts. This gives your main marble path, but it’s not quite tall enough.
7) Add a cone at the intake point, and you’ll be set!

Component B:

8) Cut an 8.5”x11” sheet of cardstock in half widthwise and roll one of the pieces along the short side, to make an 8.5” tube. Secure with tape.
9) Cut the tube at a point anywhere from 2” - 4” at about a 45 degree angle.
10) Rotate one of the pieces to form an elbow, and tape back together.
11) Cut a 1“ strip from the remaining 8.5” x 5.5” cardstock half (lengthwise) and make a bridge to use as a triangular base for the tube to sit in.
12) Use the remaining cardstock to make an input cone for the top of Component B. Trim tube and cone to get appropriate height.

Voila! Your very own Marble Run!

Note: It is highly unlikely that your students will come up with anything this clean and stable. That is OKAY! The version here is meant to be messed with and reused.

Tube

Figure 1

Ramp

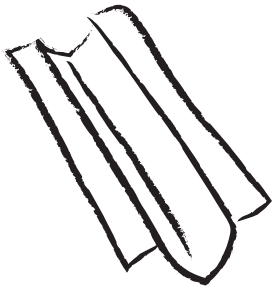


Figure 2

Bridge

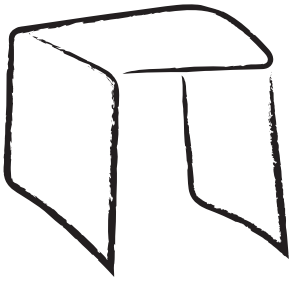


Figure 3

Cone

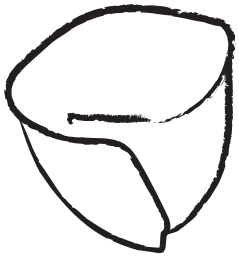
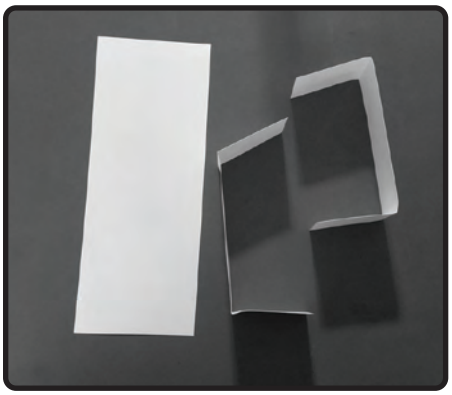


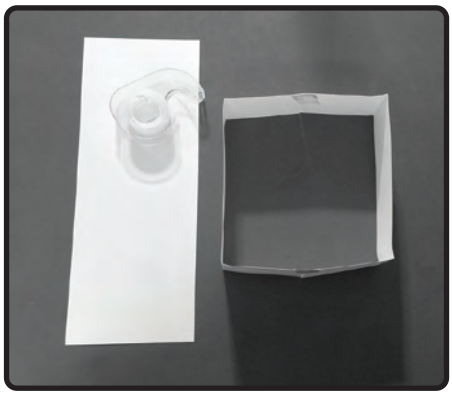
Figure 4

Marble Run

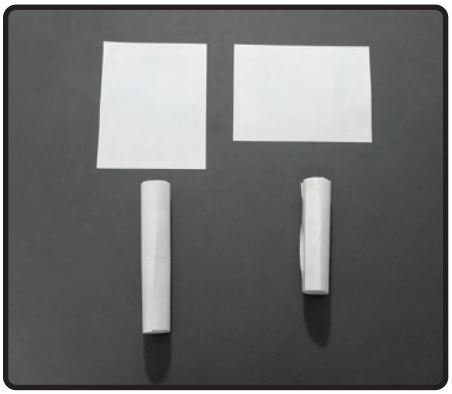
Teacher Prep Guide



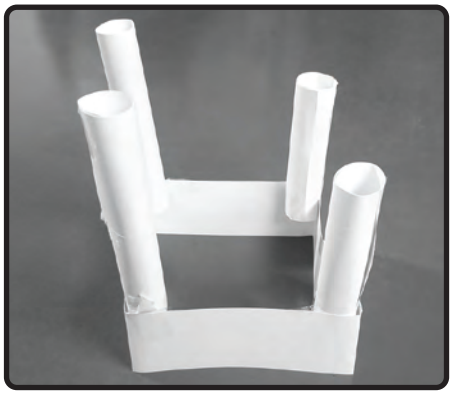
Step 1: Fold strips "bridge style"



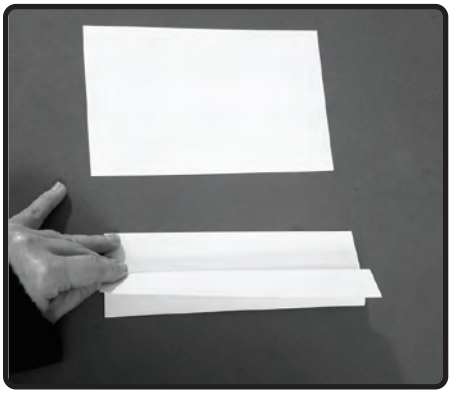
Step 2: Tape ends of folded strips together to make a base.



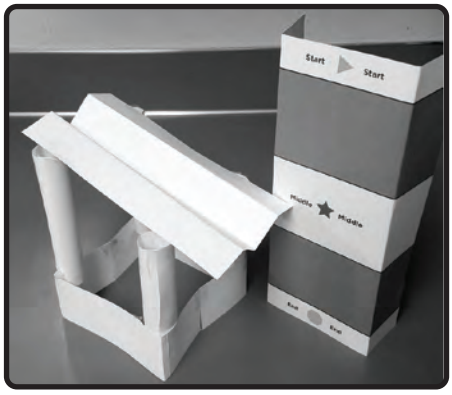
Step 3: Roll quartered paper into tubes and secure with tape.



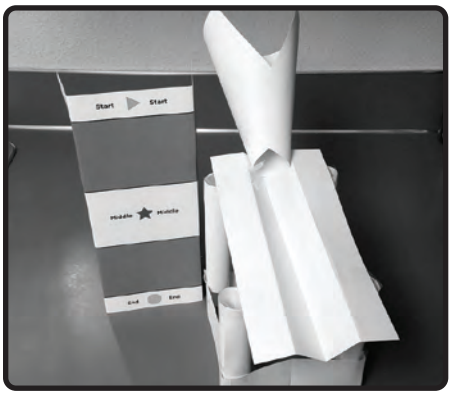
Step 4: Tape tubes inside case. Make sure to tape them near the top for height.



Step 5: Make a ramp from a half sheet of cardstock.

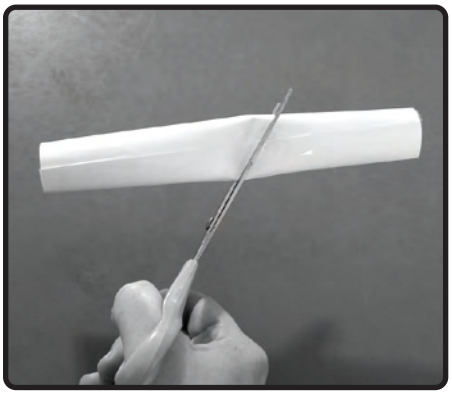


Step 6: Tape ramp to base and check height.

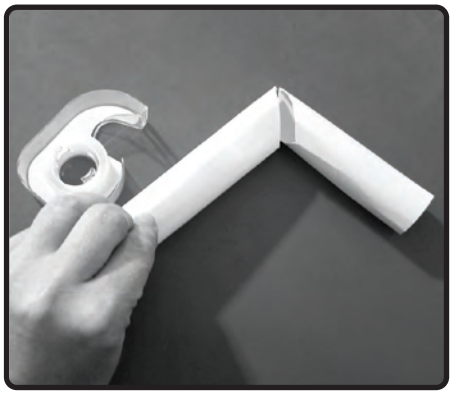


Step 7: Add cone to finish Component A.

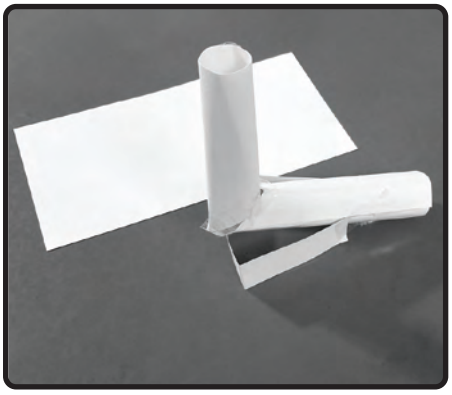
Step 8: Roll 1/2 sheet into tube to start on Component B.



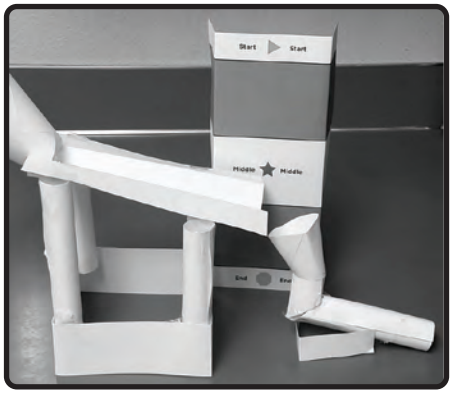
Step 9: Cut tube at an angle.



Step 10: Tape tubes back together to make an elbow.



Step 11: Make a base from a thin strip of cardstock.



Step 12: Add a cone to the top and trim pieces to size.

Marble Run Hints

Student Handout

Try using some of these:

Tube

Ramp

Bridge

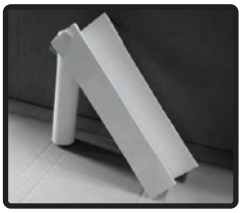
Cone

Marble Run Hints

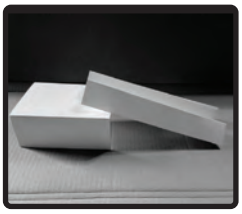
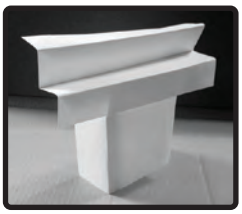
(Continued)

Now try putting them together!

Tube + **Ramp** →

 or 

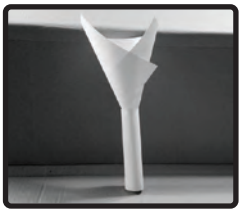
Ramp + **Bridge** →

 or 

Bridge + **Tube** →

 or 

Cone + **Tube** →

 or 

Lesson 3: Real-Life Algorithms: Plant a Seed

Unplugged | Algorithms

Overview

In this lesson, students will relate the concept of algorithms back to everyday, real-life activities by planting an actual seed. The goal here is to start building the skills to translate real-world situations to online scenarios and vice versa.

Purpose

In this lesson, students will learn that algorithms are everywhere in our daily lives. For example, it is possible to write an algorithm to plant a seed. Instead of giving vague or over-generalized instructions, students will break down a large activity into smaller and more specific commands. From these commands, students must determine a special sequence of instructions that will allow their classmate to plant a seed.

Agenda

Warm Up (10 min)

- **Vocabulary**
- **What We Do Daily**

Main Activity (20 min)

- **Real-Life Algorithms: Plant A Seed - Worksheet**

Wrap Up (10 - 20 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (15 min)

- **Real-Life Algorithms - Assessment**

Extended Learning

- **Go Figure**

Objectives

Students will be able to:

- Decompose large activities into a series of smaller events.
- Arrange sequential events into their logical order.

Preparation

- [ ] Watch the **Plant a Seed - Teacher Video**.
- [ ] Prepare supplies for planting seeds. You'll need seeds, dirt, and paper cups for each student or group.
- [ ] Print one **Real-Life Algorithms: Plant A Seed - Worksheet** for each student.
- [ ] Print one **Real-Life Algorithms - Assessment** for each student.
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **Plant a Seed** - Teacher Video
- **Real-Life Algorithms: Plant A Seed** - Worksheet
- **Real-Life Algorithms** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Algorithm** - A precise sequence of instructions for processes that can be executed by a computer

Teaching Guide

Warm Up (10 min)

Vocabulary

This lesson has one vocabulary word that is important to review:

Algorithm - Say it with me: Al-go-ri-thm

A list of steps that you can follow to finish a task

What We Do Daily

- Ask your students what they did to get ready for school this morning.
  - Write their answers on the board
  - If possible, put numbers next to their responses to indicate the order that they happen
    - If students give responses out of order, have them help you put them in some kind of logical order
    - Point out places where order matters and places where it doesn't
- Introduce students to the idea that it is possible to create algorithms for the things that we do everyday.
  - Give them a couple of examples, such as making breakfast, tying shoes, and brushing teeth.
- Let's try doing this with a new and fun activity, like planting a seed!

Main Activity (20 min)

Real-Life Algorithms: Plant A Seed - Worksheet

You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to help each other plant a seed. Directions:

- Cut out the steps for planting a seed from **Real-Life Algorithms: Plant A Seed - Worksheet**.
- Work together to choose the six correct steps from the nine total options.
- Glue the six correct steps, in order, onto a separate piece of paper.
- Trade the finished algorithm with another person or group and let them use it to plant their seed!

Lesson Tip

You know your classroom best. As the teacher, decide if you should all do this together, or if students should work in pairs or small groups.

Wrap Up (10 - 20 min)

Flash Chat: What did we learn?

- How many of you were able to follow your classmates' algorithms to plant your seeds?
- Did the exercise leave anything out?
  - What would you have added to make the algorithm even better?
  - What if the algorithm had been only one step: "Plant the seed"?
    - Would it have been easier or harder?
    - What if it were forty steps?
- What was your favorite part about that activity?

Lesson Tip

If deciding on the correct steps seems too difficult for your students, do that piece together as a class before you break up into teams.

Journaling

Ask the students to go back to their desks to reflect individually on what they learned. Write a couple of the questions up above on a whiteboard. Ask the students to discuss these in their journal. Sample prompts include:

Journal Prompts:

- Draw the seed you planted today.
- Write the algorithm you used to plant the seed.

Assessment (15 min)

Real-Life Algorithms - Assessment

- Hand out the **Real-Life Algorithms - Assessment** and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Go Figure

- Break the class up into teams.
- Have each team come up with several steps that they can think of to complete a task.
- Gather teams back together into one big group and have one team share their steps, without letting anyone know what the activity was that they had chosen.
- Allow the rest of the class to try to guess what activity the algorithm is for.

Name: _____ Date: _____

Real-Life Algorithms

Plant a Seed Worksheet

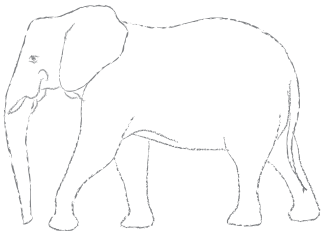
Unplugged

You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to help each other plant a seed.

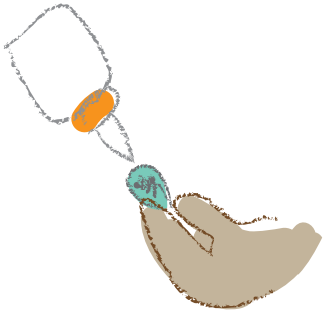
Cut out the steps of planting a seed below, then work together to glue the six the correct steps, in order, onto a separate piece of paper. Trade your finished algorithm with another person or group and let them use it to plant their seed!

PUT POT IN SUNLIGHT

PUT SEED IN HOLE



HUG AN ELEPHANT



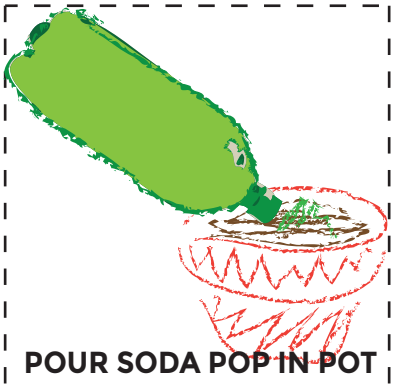
PUT GLUE ON SEED

FILL POT WITH SOIL

POKE HOLE IN SOIL

WATER POT

COVER SEED WITH SOIL



POUR SODA POP IN POT

U

Unplugged

Name: _____ Date: _____

Real-Life Algorithms

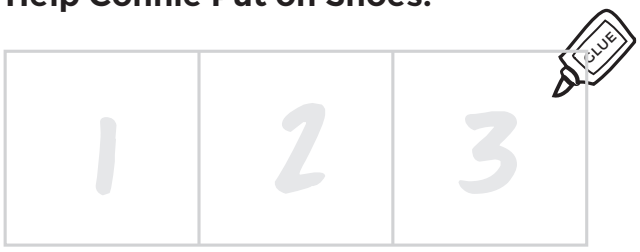
Assessment Worksheet

C O D E

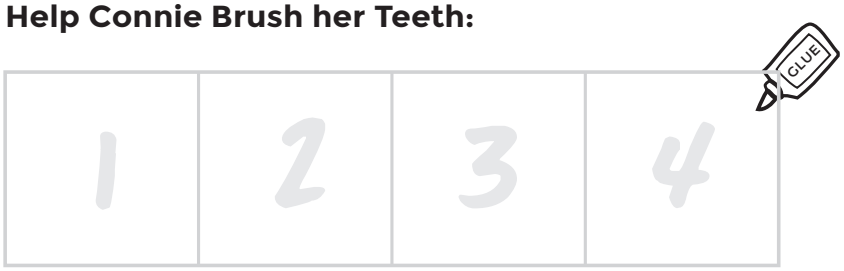
An algorithm is a list of steps that you can follow to finish a task. We follow algorithms every day when it comes to activities like making the bed, making breakfast, or even getting dressed in the morning.

Connie the Coder just woke up and is still feeling very sleepy. Can you put together some algorithms to help Connie get ready for the day?

Help Connie Put on Shoes:



Help Connie Brush her Teeth:



Help Connie Plant a Seed:

Name: _____ Date: _____

Real-Life Algorithms

Assessment Worksheet

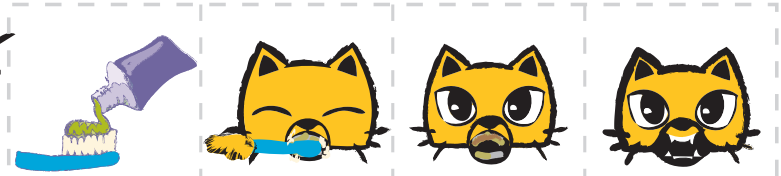
C O D E

These items are out of order. To help Connie, cut out each picture and rearrange them into the right sequence in the category above.

Putting on Shoes

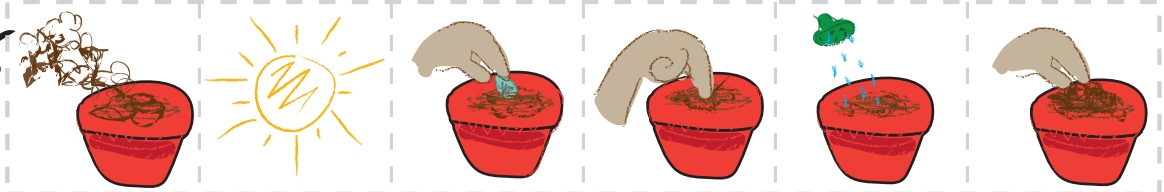
PUT FOOT IN SHOE | PICK UP SHOES | TIE SHOE

Brusing Teeth



PASTE ON BRUSH | BRUSH TEETH | DIRTY TEETH | CLEAN TEETH

Plant a Seed



FILL POT WITH SOIL | PUT POT IN SUNLIGHT | PUT SEED IN HOLE | POKE HOLE IN SOIL | WATER POT | COVER SEED WITH SOIL

Lesson 5: Programming Unplugged: Happy Maps

Unplugged | Algorithms | Sequencing

Overview

The bridge from algorithms to programming can be a short one if students understand the difference between planning out a sequence and encoding that sequence into the appropriate language. This activity will help students gain experience reading and writing in shorthand code.

Purpose

This unplugged lesson brings together teams with a simple task: get the "flurb" to the fruit. Students will practice writing precise instructions as they work to translate instructions into the symbols provided. If problems arise in the code, students should also work together to recognize bugs and build solutions.

Agenda

Warm Up (15 - 20 min)

- **Step-by-Step**
- **Discuss**
- **Vocabulary**

Main Activity (15 - 20 min)

- **Happy Maps Programming**

Wrap Up (8 min)

- **Flash Chat: What did we learn today?**
- **Journaling**

Extension Activities

Objectives

Students will be able to:

- Translate an algorithm into a program.
- Decode and run a program created by someone else.

Preparation

- [ ] Watch the **Happy Maps - Teacher Video**.
- [ ] Print out **Happy Map Cards - Worksheet** for each group.
- [ ] Print out **Happy Map Game Pieces - Manipulatives** for each group.
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **Happy Map Cards** - Worksheet (**PDF** | **DOCX**)
- **Happy Map Game Pieces** - Manipulatives (**PDF** | **DOCX**)
- **Happy Maps** - Teacher Video
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

Warm Up (15 - 20 min)

Step-by-Step

Goal: This portion of the lesson will set the stage for making the connection between an algorithm and a program.

- Ask your students for directions to the chalkboard.
  - If they start shouting simultaneously, explain that you can only hear one instruction at a time. Call on students individually if that helps.
- When you reach the board, ask for instructions to draw a smiley face.
  - Again, request this one step at a time
- Explain that many tasks can be described using a specific list of instructions. That list is called an algorithm.

This is where we introduce the activity. In your *Happy Map Cards*, there are single and double step maps. Show the students that the point of these maps is to figure out how to get the Flurb to the fruit. Students should then use their words to solve these puzzles in small groups. Example algorithms include:

- Move the Flurb up
- Move the Flurb North, then West

Discuss

- Ask if any students want to share their algorithm for one of the mazes.
  - Can everyone see how the volunteer came up with that answer?
  - Is there any debugging that needs to be done?
- Now, what if we tried to move our Flurb through a 10-step maze?
  - Could we remember all of the steps?
  - What if we had to write all of the steps down in words?
  - How could we make this easier?

Lesson Tip:

If you have time and motivation, get your students to bring their stuffies into school for this lesson and have them create programs to move stuffies from square to square, outlined in tape on the carpet.

Show students how to represent the algorithms that they just created using arrows (either drawn, or cut from the *Happy Maps Game Pieces*). Have a short discussion about how quick and easy this “code” makes the process of getting the Flurb where it needs to be.

Vocabulary

- **Program (or Code)**: an algorithm that has been encoded into something that can be run by a machine.

Main Activity (15 - 20 min)

Happy Maps Programming

Goal: Happy Maps Programming will help students translate an algorithm into code.

Now that the students have had some practice encoding algorithms, have them work on some larger maps.

Encourage the students to follow these steps:

- Discuss an algorithm to get the Flurb to the fruit.
- Encode the algorithm into arrows to share with the class.
- Try their code to see if everything works as expected.
- Debug any issues and fix their code until it works correctly.

Make sure to bring the class back together at least a couple of times to allow students to share their code or the things that they have learned.

Wrap Up (8 min)

Flash Chat: What did we learn today?

When it's time to wind down class, ask students if they can tell the difference between an algorithm and code.

Both are a list of steps, but code (a program) has been encoded in a way that can be run by a machine (or a kindergartener!)

Do you think that someone who spoke another language would be able to run your program? Why or why not?

Journaling

Journal Prompts:

Students should be encouraged to capture their thoughts in their journal after each activity (with text or images.)

Choose a journal prompt that will help students remember the purpose of this exercise. These could include:

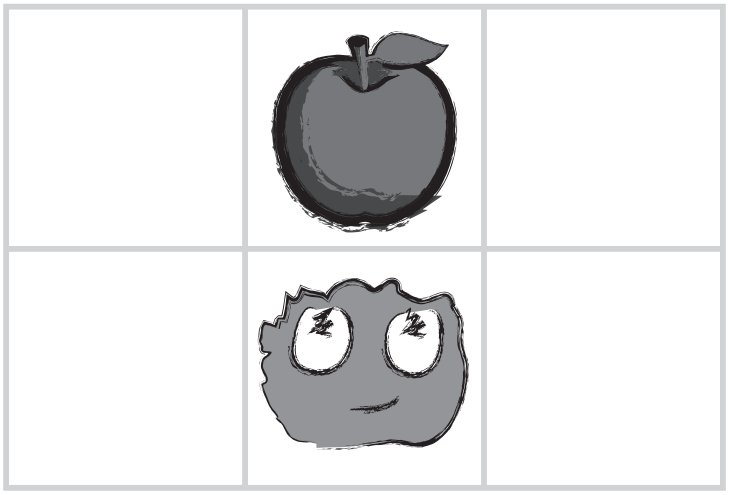
- In this game, we made programs for *people* to run. What else can read a program?
- How did you know if there was a bug in your program?

Extension Activities

- Create a life-size grid on the rug with tape and have student bring stuffies to school. Now students can program friends to move their actual stuffies as directed in the programs.
- Have students create their own maps for other students to solve using programs.

1

Happy Map 1

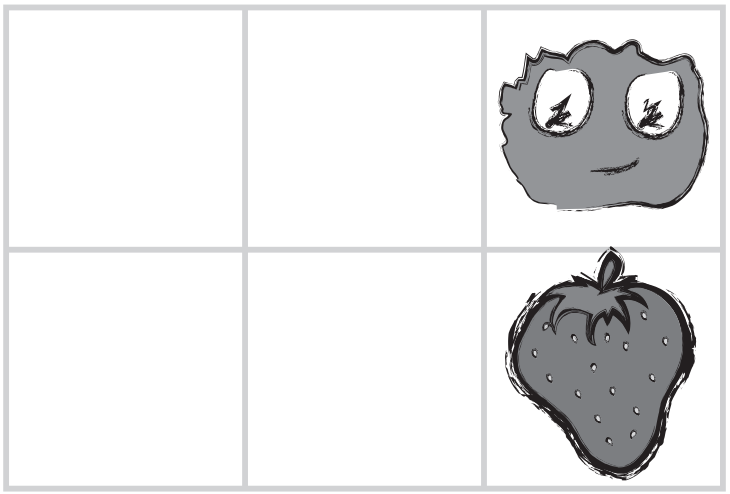


Which way should the Flurb step to get to the supplies?

Revision 161003.1a

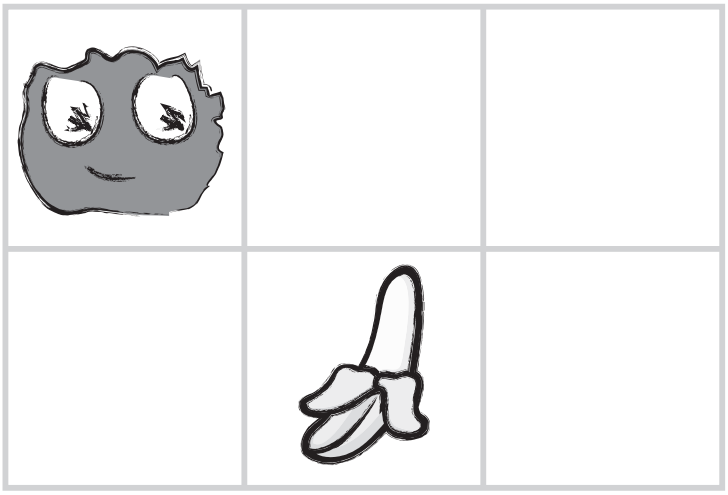
2

Happy Map 2

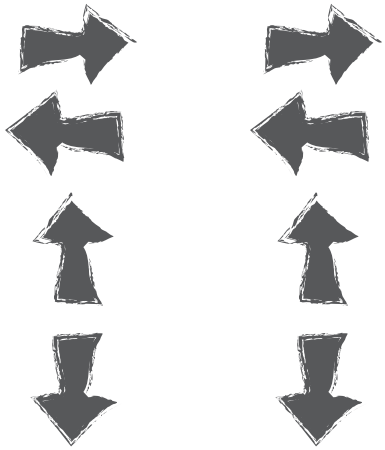


Which way should the Flurb step to get to the supplies?

3 Happy Map 3

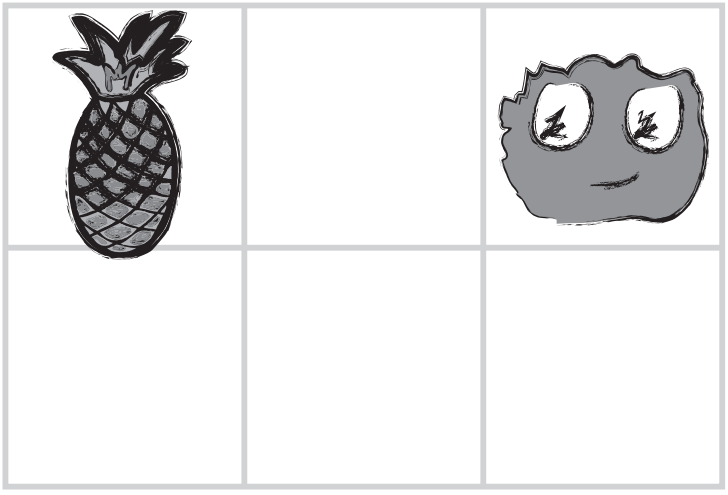


Which two ways should the Flurb step to get to the supplies?

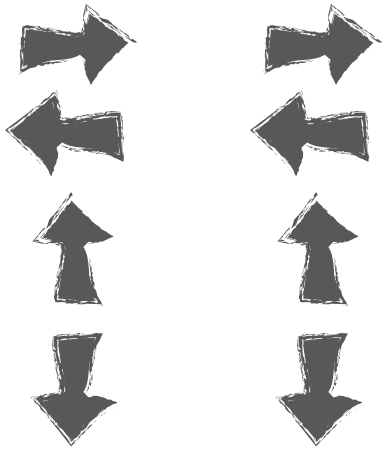


Revision 161003.1a

4 Happy Map 4

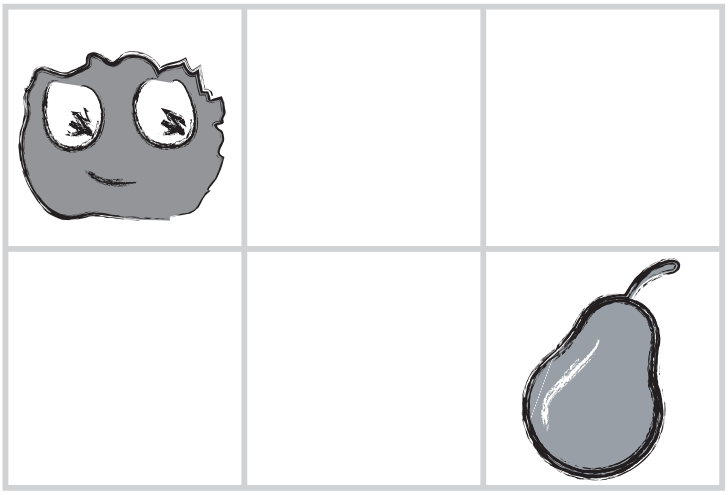


Which two ways should the Flurb step to get to the supplies?

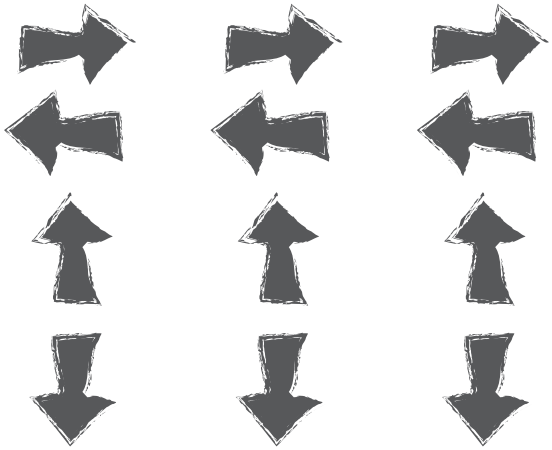


5

Happy Map 5



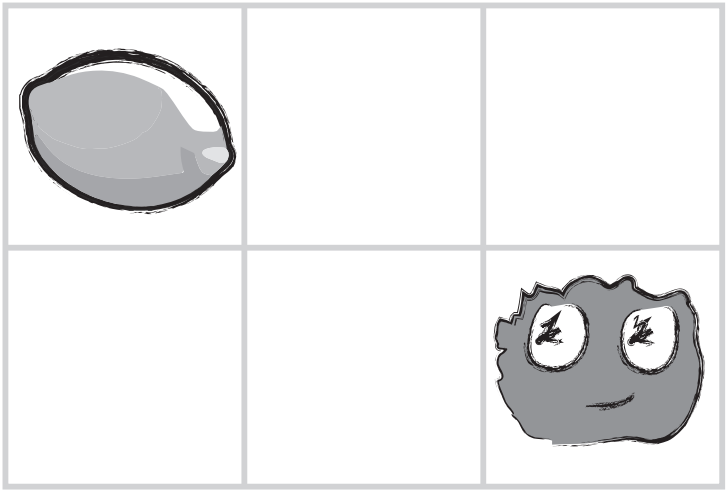
What should the Flurb do to get to the supplies?



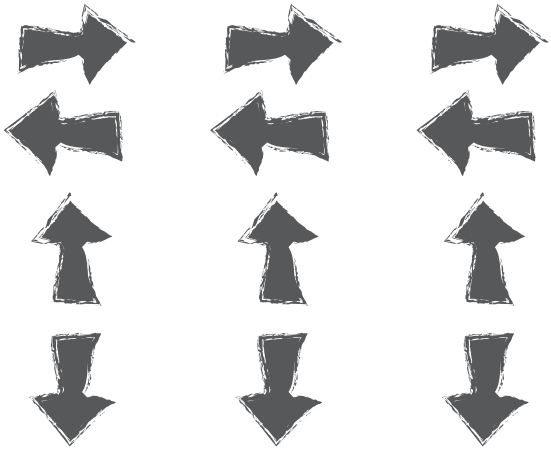
Revision 161003.1a

6

Happy Map 6



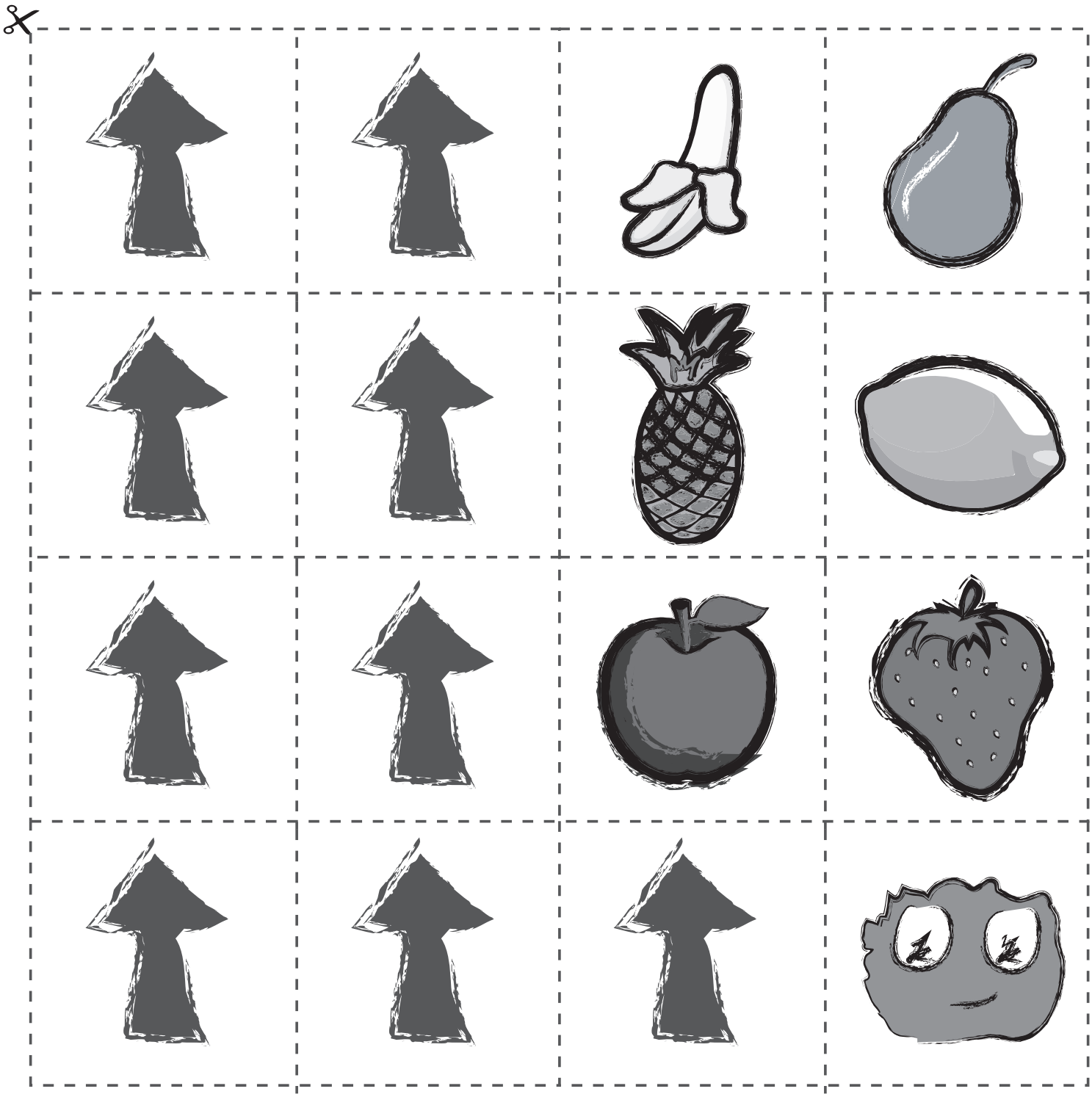
What should the Flurb do to get to the supplies?



Happy Map Blank

Name: _____ Date: _____

Happy Maps Game Pieces



Lesson 7: Common Sense Education: Going Places Safely

Common Sense Education | Unplugged

Overview

In collaboration with **Common Sense Education - Website**, this lesson helps students learn that many websites ask for information that is private and discusses how to responsibly handle such requests. Students also find out that they can go to exciting places online, but they need to follow certain rules to remain safe.

Purpose

Common Sense Education has created this lesson to teach kids the importance of being safe online. By relating places in the real world to websites on the internet, students will make important connections between safe websites and safe places in their own neighborhood.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Where We Go**

Main Activity (20 min)

- **Keep It Private**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (5 min)

- **Keep It Private - Assessment**

Extended Learning

Objectives

Students will be able to:

- Understand that being safe when they visit websites is similar to staying safe in real life
- Learn to recognize websites that are safe for them to visit.
- Recognize the kind of information that is private and understand that it should never be shared online.

Preparation

- [ ] Watch the **Going Places Safely - Teacher Video**.
- [ ] Prepare to show the **Going Places Safely - Lesson Video**.
- [ ] Live access or print-off of **SecretBuilders** sign-up page (Click “New Player,” select an age, and then select “I’m a Girl” or “I’m a Boy.”).
- [ ] Print one **Keep It Private - Assessment** for each student.
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **Going Places Safely** - Teacher Video
- **Going Places Safely** - Lesson Video
- **Keep It Private** - Assessment
- **Common Sense Education** - Website
- **Think Spot Journal** (**PDF** | **DOCX**)

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

Username - Say it with me: Yews-er-naym

A name you make up so that you can see or do things on a website, sometimes called a “screen name”

Where We Go

- Invite students to talk about places they have visited on a class field trip.
  - If students have limited experience with field trips, provide some examples of the types of places they could visit as a class, such as museums, science centers, or zoos.
  - Have students choose a place they would like to go on a class field trip.
- Have students take an imaginary field trip to their chosen place.
  - Narrate the preparations while having students pantomime what’s happening – For example: put on your jacket; climb on/off the bus; get your ticket checked; go inside.
    - Have students describe what they think they might see and do once they arrive.
- Let the students sit back down, then ask: "What do you need to do to stay safe when you visit new places?"

Play **Going Places Safely - Lesson Video**.

What three rules does Jeremiah follow when he goes places online?

1) Always ask your parent (or teacher) first

2) Only talk to people you know

3) Stick to places that are just right for you

Now, let's see what more we can do to keep ourselves safe.

Main Activity (20 min)

common sense education®

Keep It Private

Access **SecretBuilders** sign-up page live, or project a print-out on the board for the class to see.

- Invite students to give examples of information that they should keep private.
  - Write down their responses on the board or chart paper so that you can return to them later in the lesson.
- Make sure they understand that private information includes the following:
  - full name
  - age
  - address
  - telephone number
  - email address (or parents’ email addresses)
  - where they go to school or after school
  - where their parents work
- Encourage students to discuss why it is important to keep this information private.
  - Stress that it is never safe to give out private information to people they don’t know.

    - Students should always ask a parent or caregiver before they give out private information to anyone.
- Refer back to the sign-up page.
    - Ask "Do you think you should use your real name, or something that includes your real name, when you make up a username?"

Guide students through the following rules and tips for creating usernames:

Rules:

- Ask a parent or other trusted adult before you create a username.
- Never include any private information in your username, such as your real name, age, birthday, the name of your school or hometown, parts of your address or phone number, or email address.
- Avoid using symbols or spaces, as they are usually not allowed in usernames.

Tips

- Include the name of something that will help you remember your username, like your favorite animal, character, or toy. You might have to combine this with other words or numbers.
- If the username you create is already taken, you will have to come up with another one.
- Write down your username and password and, with the help of a parent, find a safe place to keep it in case you forget them.

Distribute paper and place students in pairs.

Directions:

1. Have students interview their partner using the following questions, and write down their responses: - What is your favorite pet or animal? - What is your favorite TV show, book, or movie character? - What are your favorite numbers?
2. Instruct students to make up three safe usernames for their partner using information from their interview responses. - They should not include their partner's name, age, school, email address, birthday, or any other private information.
3. Invite students to share one or more of their usernames with the class.
4. Encourage students to respond to one another's usernames, confirming that each name follows the rules they have learned.

Lesson Tip

For more in-depth modules, you can find additions to this curriculum at the **Common Sense Education - Website** page on Scope and Sequence.

Wrap Up (15 min)

Flash Chat: What did we learn?

- What information should you always keep private when you are using the computer?
- What rules should you follow when you make up a username?
- What can the Internet be used for?
- What rules do we have for visiting places online?

Take the time to discuss again what is appropriate information to share on the Internet, and what is not:

Appropriate	Not Appropriate
Interests	Address
Hobbies	Full Name
First Name	Information that would hurt others

Lesson Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw some things that you should never talk to a stranger about on the internet. For example, draw your house to represent your address, draw your school, or draw your family.

Assessment (5 min)

Keep It Private - Assessment

- Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Common Sense Education

- Visit **Common Sense Education - Website** to learn more about how you can keep your students safe in this digital age.

Name: _____ Date: _____

U

Unplugged

Keep It Private

Learning to be Safe and Responsible Assessment

C O D E

Just because you **can** share something online doesn't mean that you **should**!

---

1) Circle the place you would most like to visit online

THE JUNGLE

OUTER SPACE

THE OCEAN

---

2) Can you spot the private information? Mark "X" through the information that you should not share with people you do not know well.

My address is 2524 Sycamore Lane.

My birth date is February 5th, 2006

I like watermelon.

I like swimming.

---

3) On the back of this paper, draw something that you enjoy and want to share on the Internet.

Lesson 8: Loops Unplugged: Happy Loops

Unplugged | Loop | Repeat

Overview

Loops are a very helpful and powerful tool in programming. To understand how helpful loops can be, students will need to be driven to want an easier way to solve mundane problems.

Purpose

This lesson serves as an introduction to loops. *Loops* allow for students to simplify their code by grouping commands that need to be repeated. Students will develop critical thinking skills by noticing repetition in movements of their classmates and determining how many times to repeat the commands inside of the loops. By seeing 'Happy Maps' again, students will be able to relate old concepts such as sequencing to the new concept, loops.

Agenda

Warm Up (10 - 15 min)

- **Such a Long Walk**
- **Discuss**
- **Vocabulary**

Main Activity (15 - 20 min)

- **Happy Loops**

Wrap Up (8 min)

- **Journaling**

Extension Activities

Objectives

Students will be able to:

- Identify repetitive code and convert a series of multiple actions into a single loop.
- Decode loops into a series of multiple actions.

Preparation

- [ ] Print out **Happy Map Cards - Worksheet** for each group
- [ ] Print out **Happy Map Game Pieces - Manipulatives** for each group
- [ ] Print out **Happy Map Cards XL - Worksheet** for each group
- [ ] Print out **Happy Map Game Pieces Bonus Pack - Manipulatives** for each group
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **Happy Map Cards** - Worksheet (**PDF** | **DOCX**)
- **Happy Map Cards XL** - Worksheet
- **Happy Map Game Pieces** - Manipulatives (**PDF** | **DOCX**)
- **Happy Map Game Pieces Bonus Pack** - Manipulatives
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again

Teaching Guide

Warm Up (10 - 15 min)

Such a Long Walk

Goal: This portion of the lesson should help students see that there is an easier way to handle repetitive code than to brute force a solution with dozens of the same symbols.

Recall the 'Happy Maps Programming' activity with your students. Recall the limited pieces that they had to work with (up, down, left, right). Then, pull up one of the new -- and much longer -- *Happy Maps XL*.

Can your students help you program these maps? It takes a while, doesn't it? What can you do if you run out of arrow pieces?

Discuss

Give students the opportunity to brainstorm shorter ways to relay the code that they're creating. (This bit can be skipped over if your students start saying things like: "Move forward 6 times." Since that will open the discussion about how to show "six times" with symbols.)

Once students have put together the idea of "repeating" code, give them the vocabulary around it. Make sure to share with them that often the terms "repeat something" and "loop something" will be used interchangeably in Code Studio.

Vocabulary

- **Loop**: the act of doing something over and over and over.
- **Repeat**: doing something again.

Notice: Loops *repeat* a step over and over again.

Main Activity (15 - 20 min)

Happy Loops

Now that students are familiar with the ability to repeat lots of code using a single loop, select an XL map and let them help you code the situation. Do this as many times together as a class as you need, then set students off in groups to solve some problems on their own. You will also need to add the Happy Maps Game Pieces Bonus Pack to adapt this activity for loops.

Make sure to walk around and have students run through their code with you watching. Are there any bugs? Use the debugging questions to help them find a solution.

- What does it do?
- What is it supposed to do?
- What does that tell you?
- Does it work at the first step?
- Does it work at the second step?
- Where does it stop working

Wrap Up (8 min)

Journaling

Allow students to reflect on the activity that they just experienced.

Journal Prompts:

- Ask students to draw a feeling face in the corner of their journal page to remind them how they felt about this lesson.

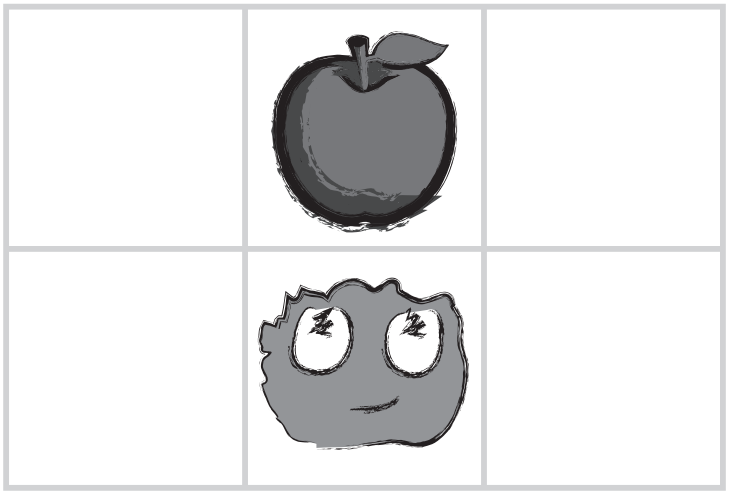
- Have the students write or draw something in their journal that will remind them later what loops are. Prompts include:
  - What does "repeat" mean to you?
  - Draw a picture of you repeating something.

Extension Activities

- Create a life-size grid on the rug with tape and have student bring stuffies to school. Now students can program friends to move their actual stuffies as directed in the programs.
- Have students create their own maps for other students to solve using loops.
- Draw a program on the board that uses several sets of repeated commands and have students take turns coming to the front to swap symbols for repeat loops.

1

Happy Map 1

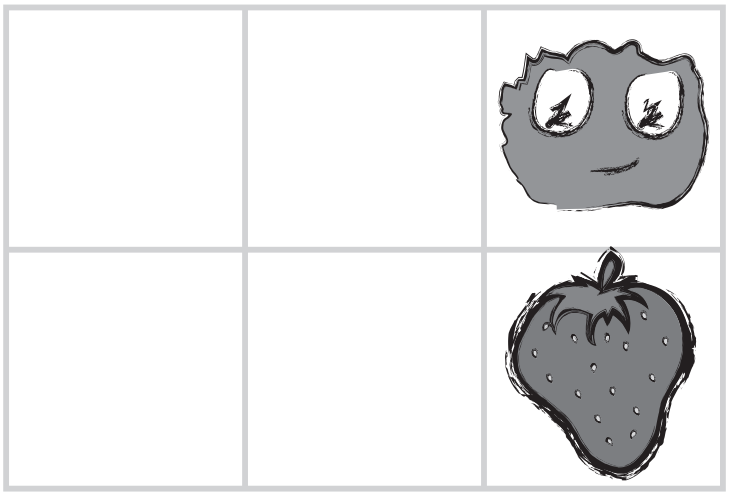


Which way should the Flurb step to get to the supplies?

Revision 161003.1a

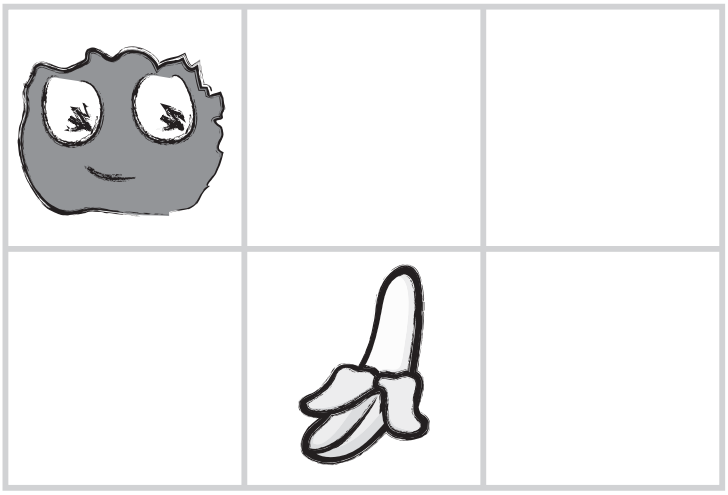
2

Happy Map 2

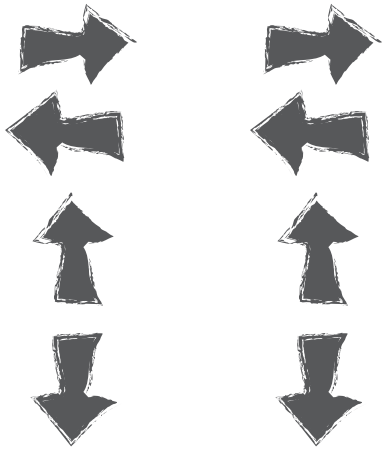


Which way should the Flurb step to get to the supplies?

3 Happy Map 3

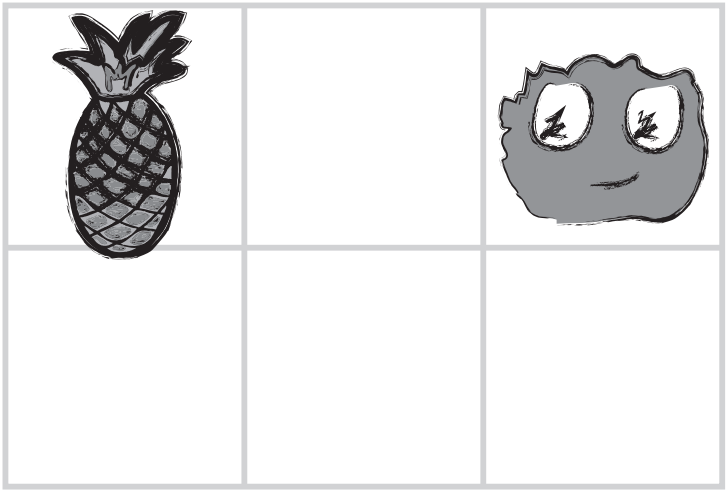


Which two ways should the Flurb step to get to the supplies?

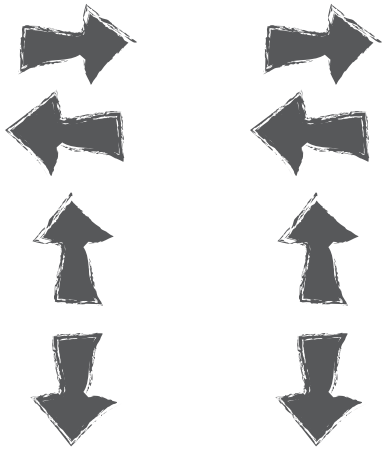


Revision 161003.1a

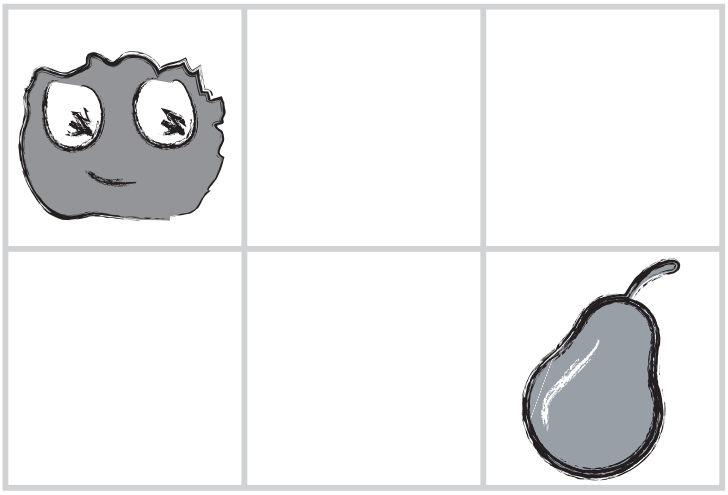
4 Happy Map 4



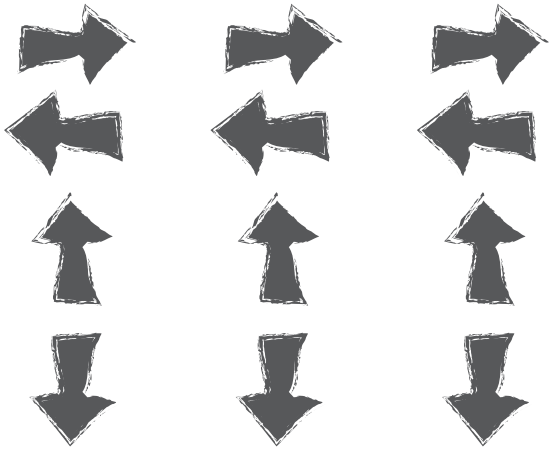
Which two ways should the Flurb step to get to the supplies?



5 Happy Map 5

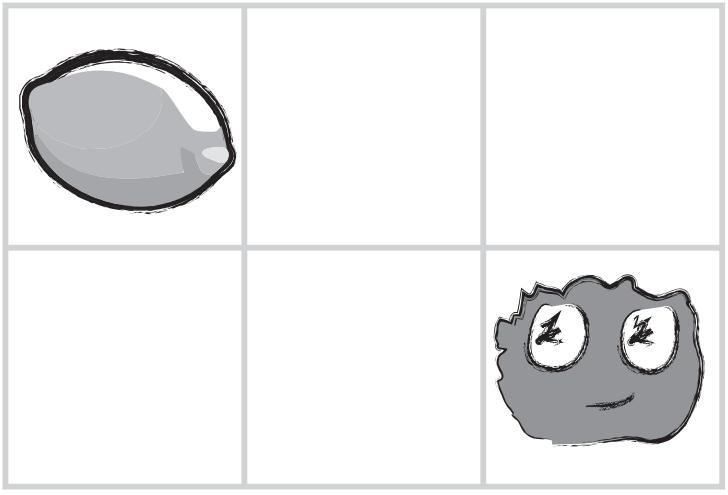


What should the Flurb do to get to the supplies?

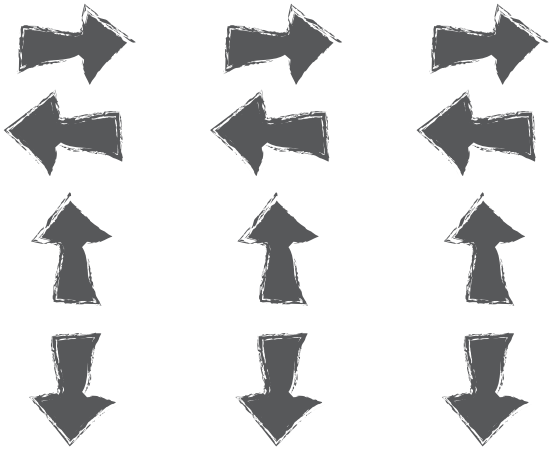


Revision 161003.1a

6 Happy Map 6



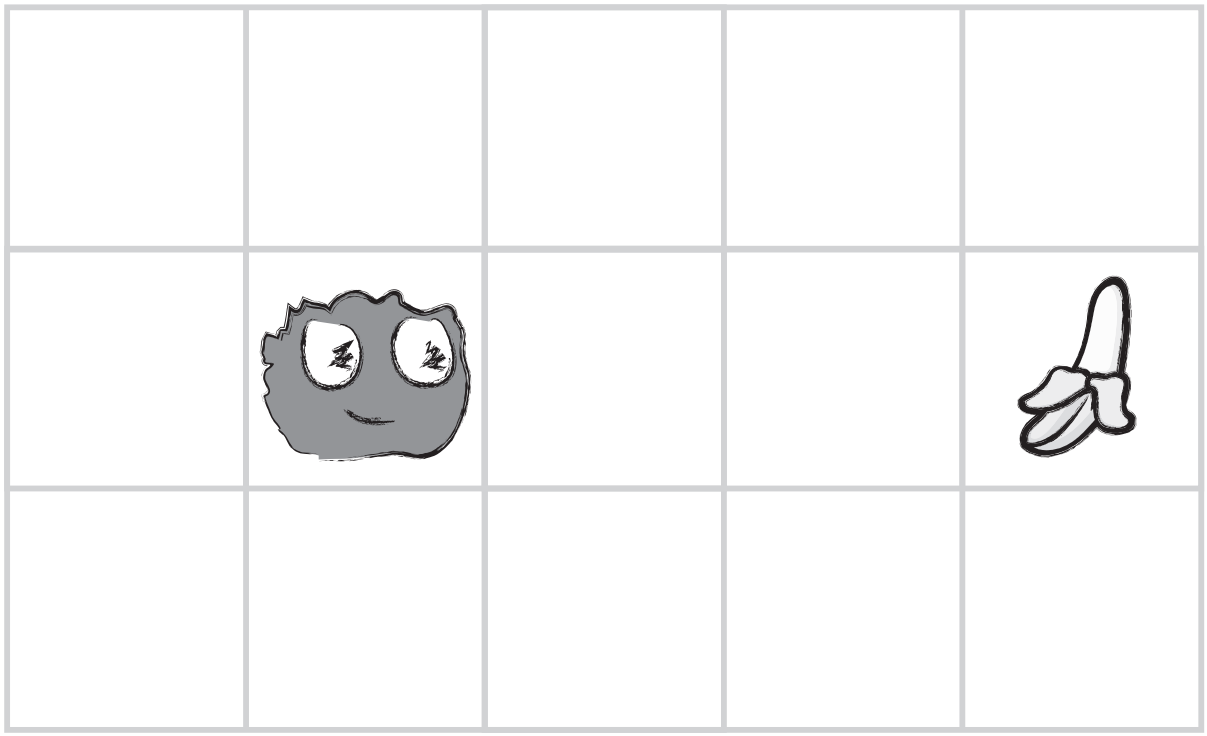
What should the Flurb do to get to the supplies?



Happy Map Blank

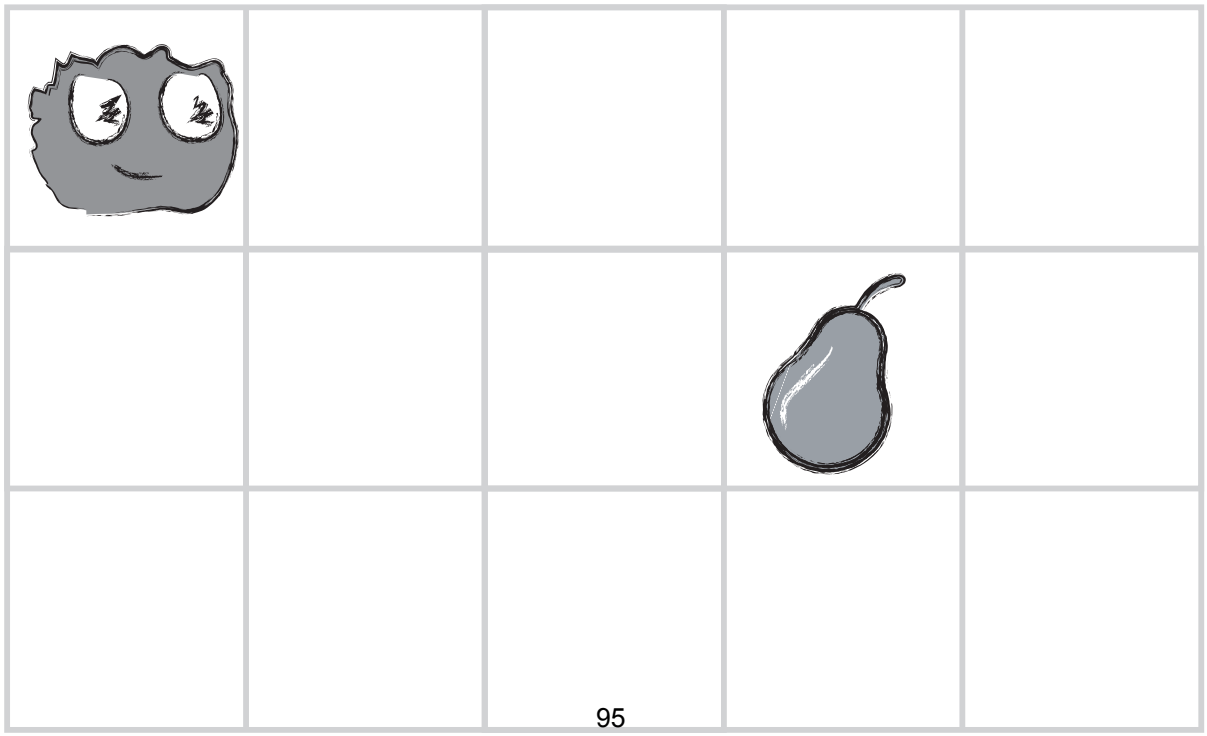
1

Happy Map XL 1

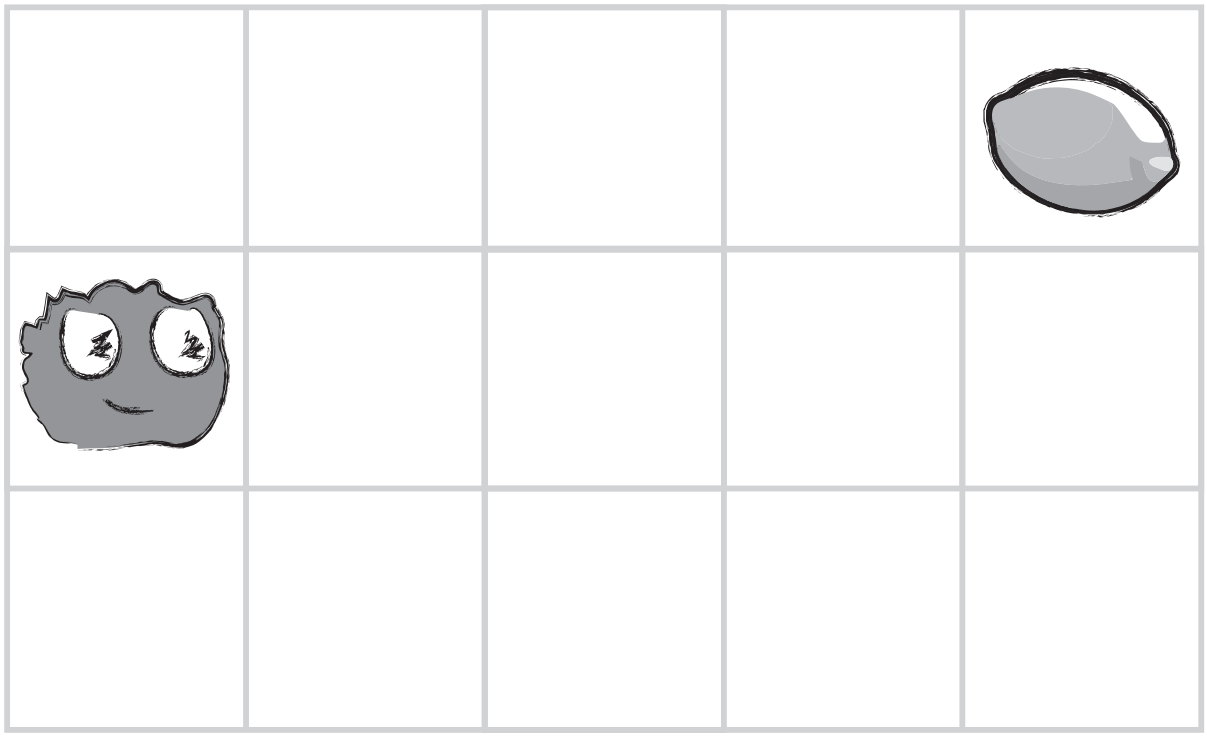


2

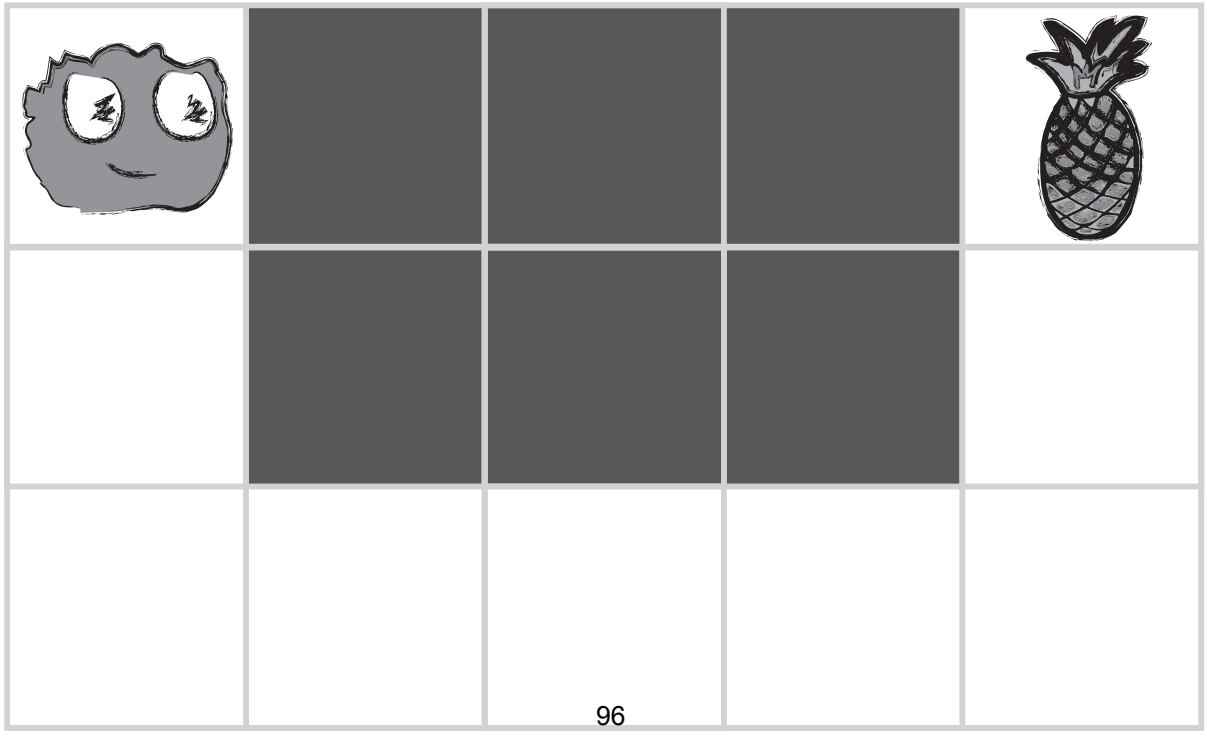
Happy Map XL 2



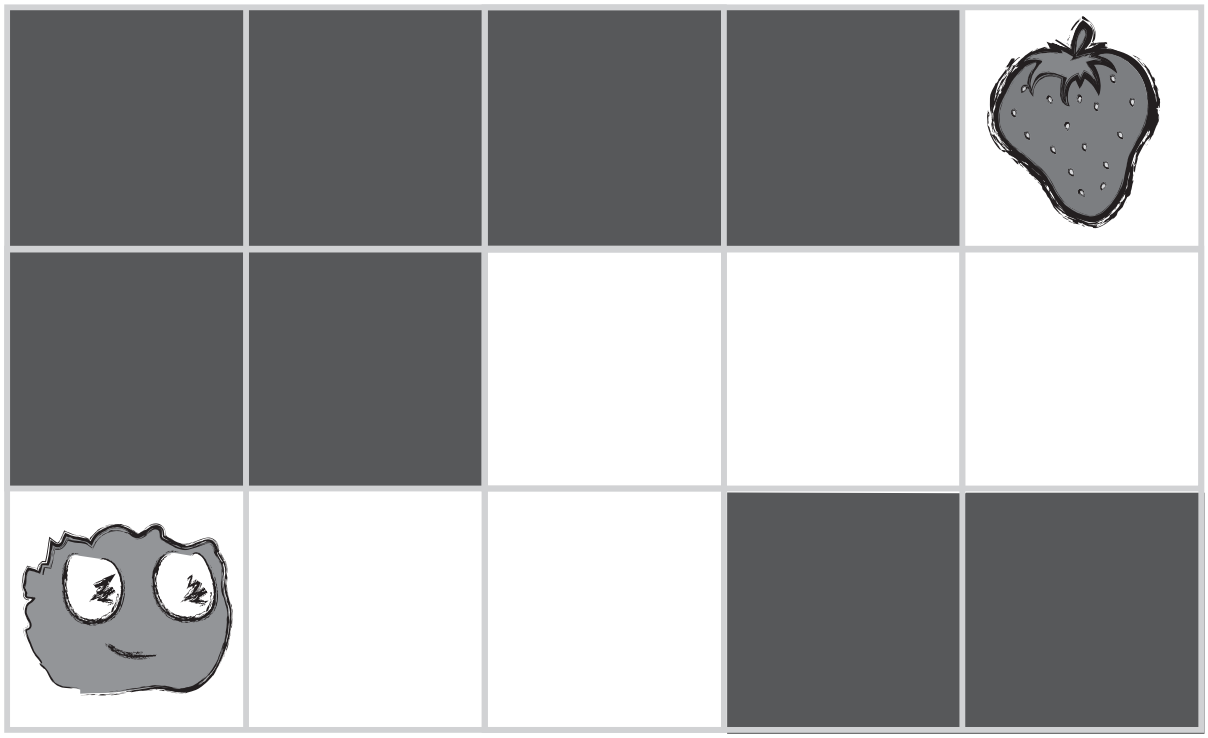
3 Happy Map XL 3



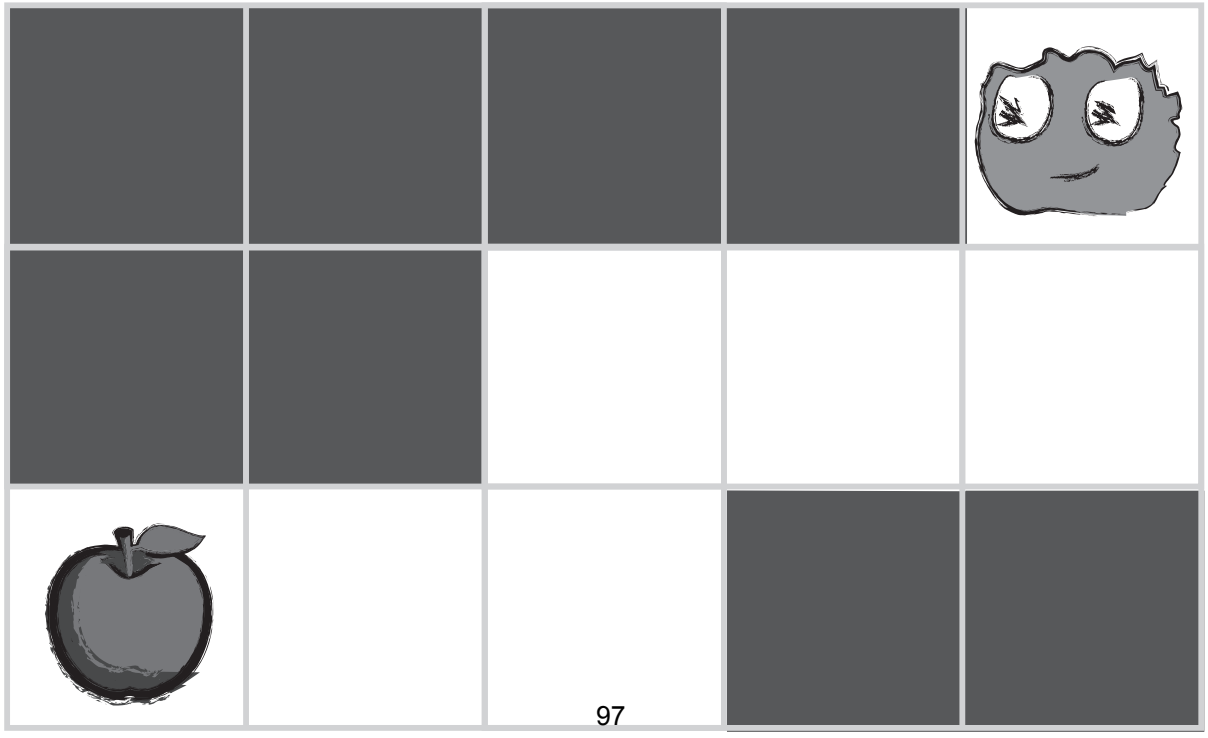
4 Happy Map XL 4



5 Happy Map XL 5



6 Happy Map XL 6



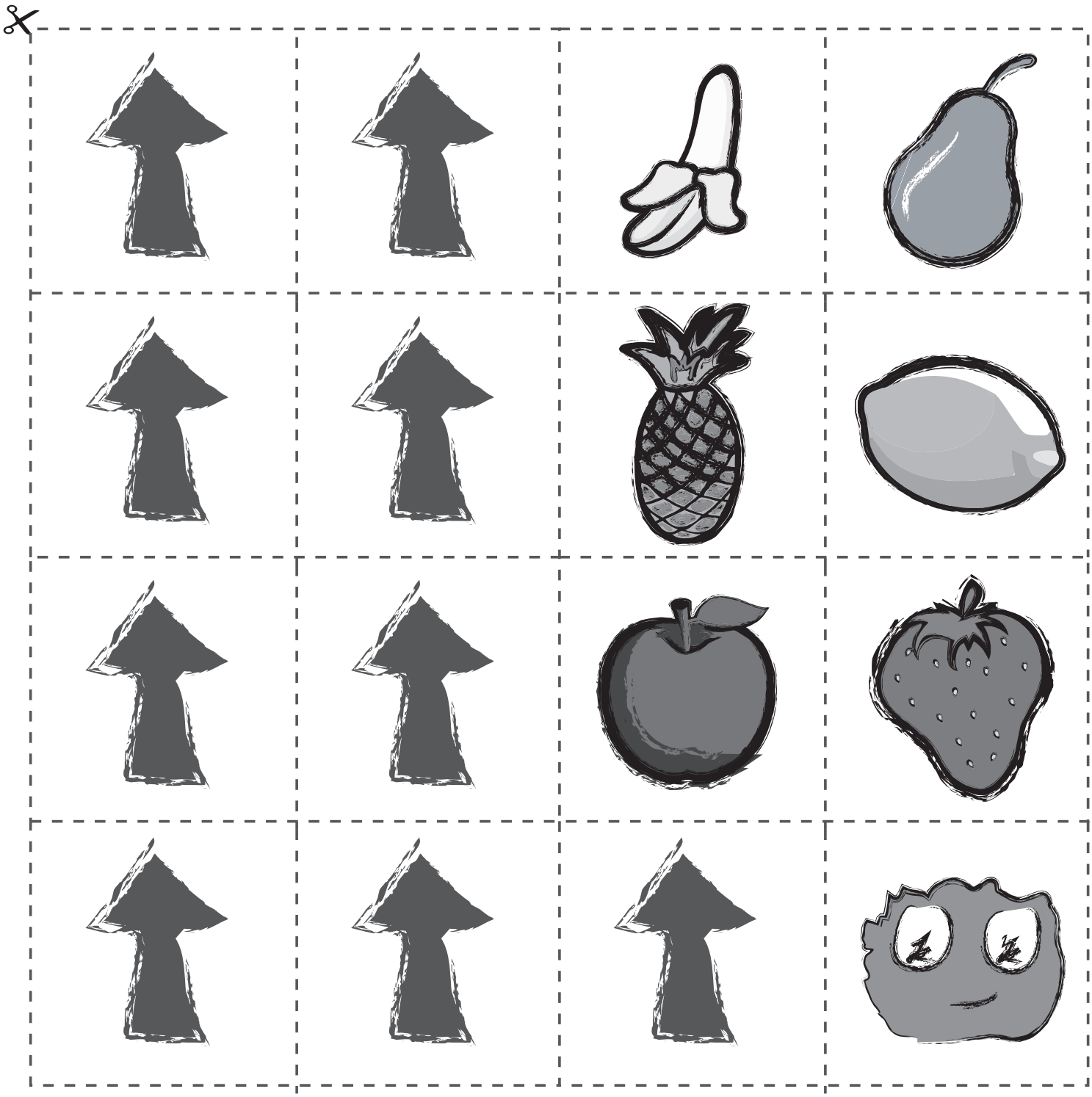
5

Happy Map XL Blank (left)

Happy Map XL Blank (right)

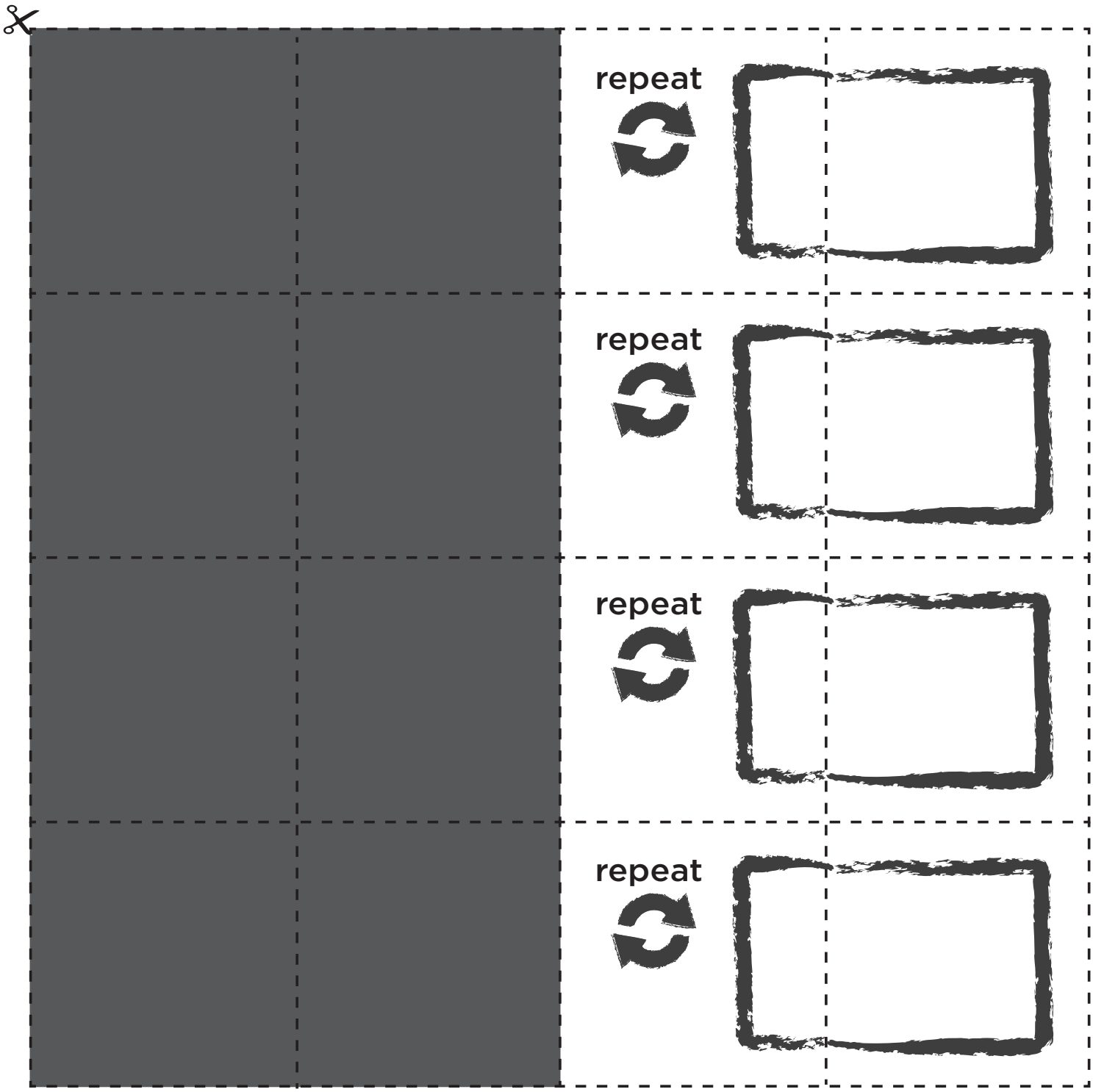
Name: _____ Date: _____

Happy Maps Game Pieces



Name: _____ Date: _____

Happy Maps
Game Pieces Bonus Pack



Lesson 11: Events Unplugged: The Big Event

Unplugged | Event

Overview

Events are a great way to add variety to a pre-written algorithm. Sometimes you want your program to be able to respond to the user exactly when the user wants it to. That is what events are for.

Purpose

Today, students will learn to distinguish events from actions. The students will see activities interrupted by having a 'button' pressed on a paper remote. When seeing this *event*, the class will react with a unique action. Events are widely used in programming and should be easily recognizable after this lesson.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **A Series of Events**

Main Activity (15 min)

- **The Big Event**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (10 min)

- **CSF The Big Event Activity - Assessment**

Extended Learning

Objectives

Students will be able to:

- Recognize actions of the teacher as signals to initiate commands.
- Practice differentiating pre-defined actions and event-driven ones.

Preparation

- [ ] Watch the **The Big Event - Teacher Video**.
- [ ] Print one **CSF The Big Event Activity - Worksheet**.
- [ ] Print **CSF The Big Event Activity - Assessment** for each student.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **The Big Event** - Teacher Video
- **CSF The Big Event Activity** - Worksheet
- **CSF The Big Event Activity** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Event** - An action that causes something to happen.

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important vocabulary word:

- **Event** - Say it with me: E-vent

An action that causes something to happen.

A Series of Events

- Prep your class to answer a question:
  - "I'm going to ask you a question. I want you to raise your hand if you want me to call on you for the answer."
  - Ask a simple question that most of your students should be able to answer, such as:
    - How many thumbs do I have?
    - What is bigger, a bird or a horse?
  - Call on a student who has their hand raised and let them give their answer.
  - Upon finishing that display, ask the class how you knew that the student wanted you to call on them.
    - Your class will likely mention the raising of the hand.
  - Explain to everyone that when students raise their hand, it is an "event" that causes you to know that they want to be called on.
- Ask the class if they can think of any other events that give signals.
  - You may need to remind them that you're not talking about an event like a birthday party or a field trip.
  - If they have trouble, you can remind them that an event is an action that causes something to happen.
    - What about an alarm clock going off? What does that make happen?
    - What about pressing "Start" on the microwave? What does that do?
    - What about pressing the power button on your tv remote?
- Today, we're going to create programs with events.

Main Activity (15 min)

The Big Event

- Do you remember helping the Flurbs find fruit?
  - In that exercise, you knew in advance exactly where you wanted your Flurb to end up, so you could make a program that took them from start to finish without any interruptions.
  - In most real programs, we can't do that because we want to have options, depending on what the user needs.
    - Say that I only want my character to move when my finger is on the screen of my phone. I would need to program the character to only move when I put my finger on the screen of my phone.
    - Putting my finger on the screen would then become an "event" that tells my character to move.

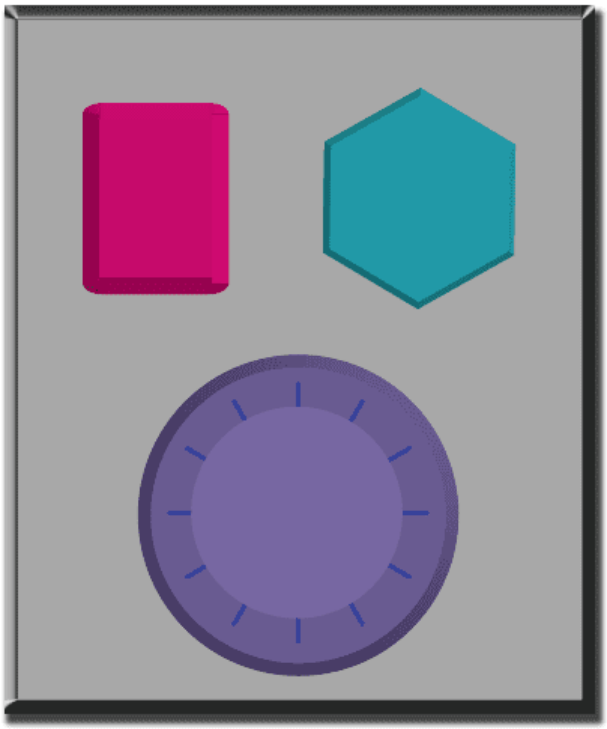
Lesson Tip

If your students seem confused, talk about their favorite games and all of the ways that they let the characters know what they're supposed to do. Point out how the game would be really boring if it ran from start to finish without any events required.

In earlier lessons, we created algorithms that allowed us to control a friend or Flurb for several steps at a time. It was fun and useful, but what happens when you don't know everything that you want your friend to do in advance? This is where events come in!

Directions:

- Project the Event Controller onto your classroom screen.



- Decide with your class what each button does. We suggest:
  - Pink Button -> Say “Wooooo!”
  - Teal Button -> “Yeah!”
  - Purple Dial -> “Boom!”
- Practice tapping the buttons on the overhead and having your class react.
- Add some button sequences into the mix and have the students try to keep up with their sounds.
- Let your class know that every time you push a button, it is an “event” that lets them know what they are expected to do next.
- Get the class started on a planned task before interrupting them again with the buttons. We suggest:
  - Counting to 10
  - Singing “Old MacDonald”
- Once their plan is underway, interject button presses sporadically.
- Continue the blend until they understand the difference between actions that are guided by a plan and those that are event driven.

Wrap Up (15 min)

Flash Chat: What did we learn?

- Why do we need to be able to handle events in a program?
- What are some other kinds of events that you can think of?

Journaling

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- Draw an event that caused an action today.
- Draw an action that was caused by an event that happened today.

Assessment (10 min)

CSF The Big Event Activity - Assessment

- Hand out the assessment activity and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

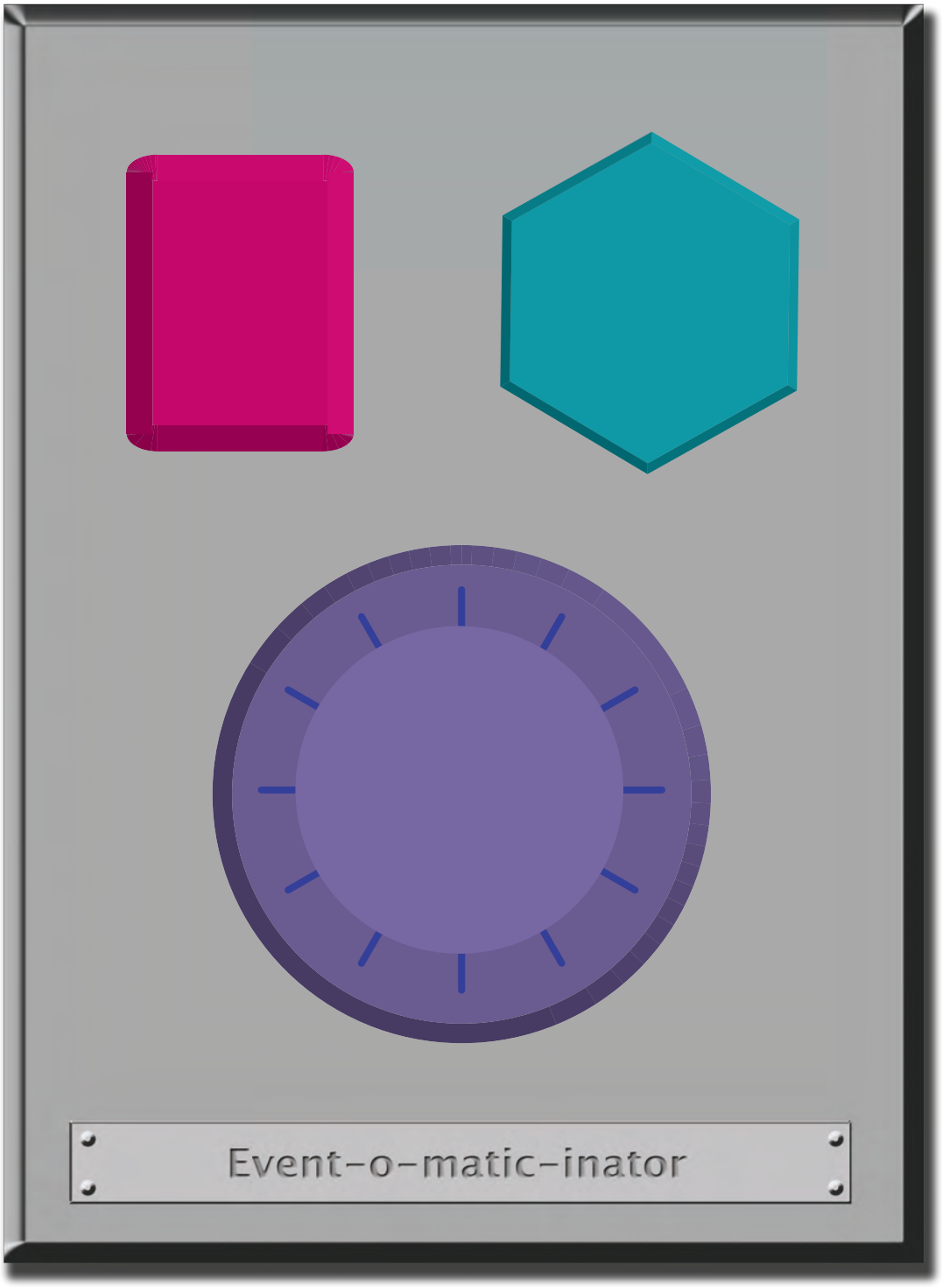
Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

One Person's Event is Another One's Reaction

Assign each student an event to watch out for, and an appropriate reaction to that event. Chain the actions so that each child's reaction becomes an event that triggers the reaction of another student. Keep assigning until everyone has something to do and everyone makes someone react.

Eventopalooza

Break the class up into groups. Using the Events Controller, assign each group a different reaction to the same button. Do this for all three buttons, then watch the chaos!


Event-o-matic-inator

U

Unplugged

Name: _____ Date: _____

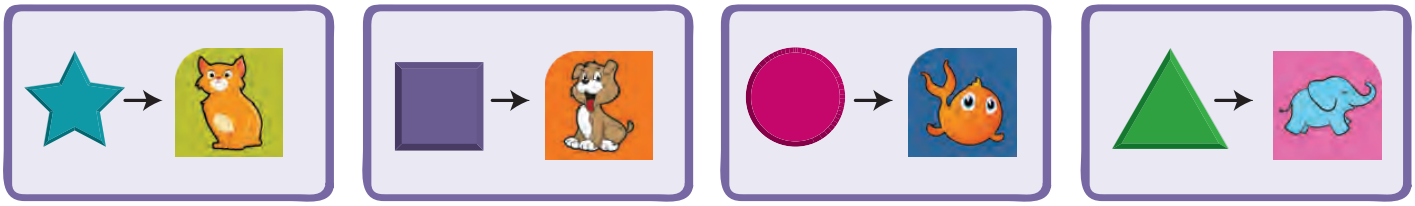
The Big Event

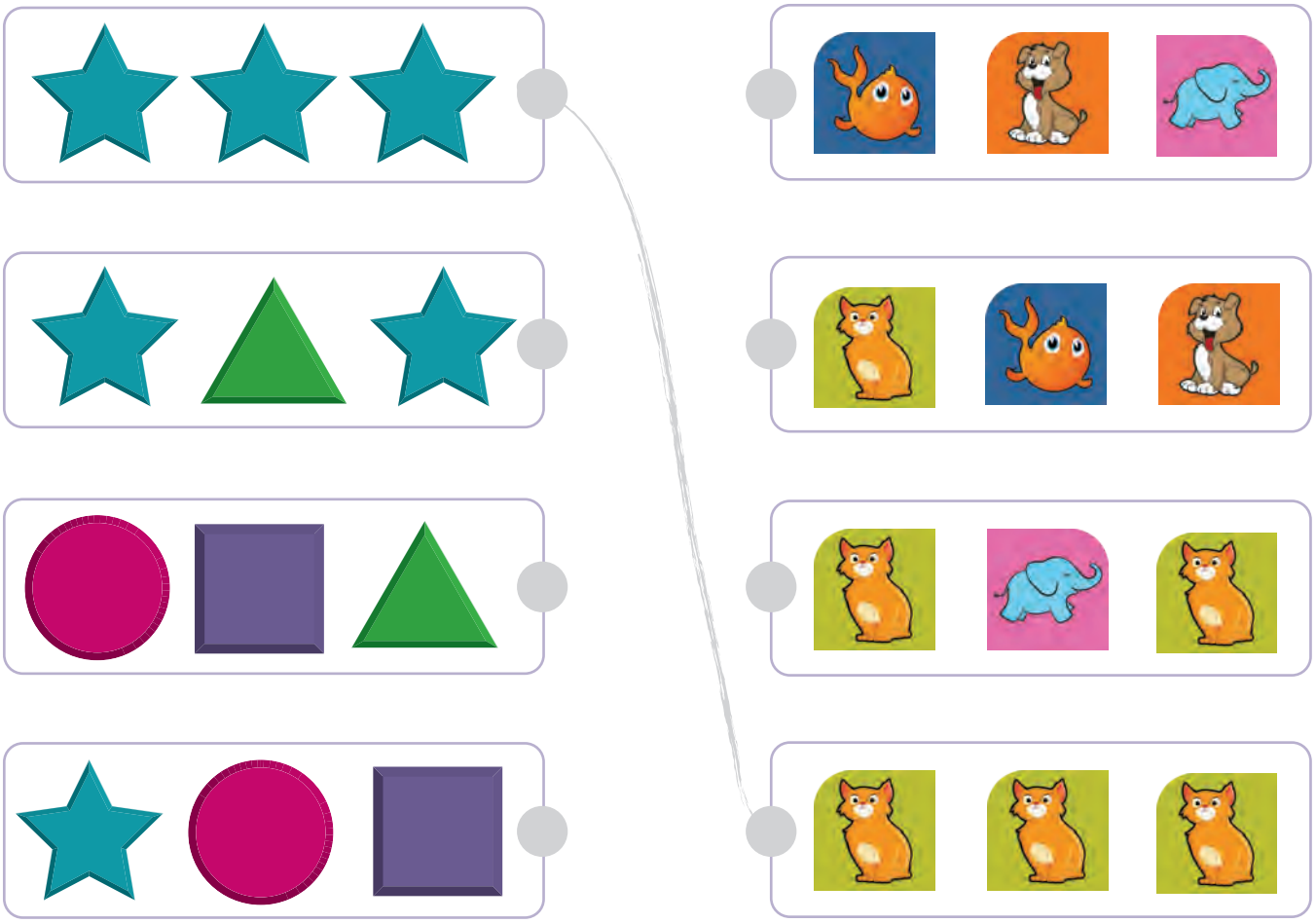
Controlling by Events Assessment

C O D E

You've been given a magical controller that changes the picture on the frame on your desk.

Take a look below to see what each button does. Can you figure out which series of button events will cause your frame to show the pictures on the right? Draw a line from each set of pictures to the button combination that causes it. The first one has been done for you.





Course B

Lesson 1: Debugging: Unspotted Bugs

Unplugged | Bug | Debugging | Persistence

Overview

This lesson will guide students through the steps of debugging. Students will learn the mantra: "What happened? What was supposed to happen? What does that tell you?"

Purpose

Research shows that some students have less trouble debugging a program than writing one when they first learn to code. In this lesson, we introduce the idea of debugging in a real world sense.

The goal in this lesson is to teach students steps to spot a bug and to increase persistence by showing them that it's normal to find mistakes. In later lessons, students will debug actual programs on Code.org.

Agenda

Warm Up (12 min)

- **Unspotted Bugs**
- **Vocabulary**

Marble Run Breakdown (10 - 20 min)

- **Debug the Run**

Wrap Up (10 - 20 min)

- **Journaling**

Extended Learning

- **Real Life Bug Hunting**

Objectives

Students will be able to:

- Express that they have noticed when something goes differently than what is expected.
- Identify what the expected result was before an error occurs.
- Determine and describe the difference between what was expected and what actually happened in the event of an error.

Preparation

- [ ] Review the Unspotted Bugs Story (**Unspotted Bugs - Online Story**).
- [ ] Pre-read Unspotted Bugs to identify appropriate questions for your classroom.
- [ ] Follow instructions in the **Marble Run - Teacher Prep Guide** to make a Marble Run (which will be arranged incorrectly at the start).
- [ ] Give a **Think Spot Journal** to each student.

Links

For the Teacher

- **Marble Run** - Teacher Prep Guide (**PDF** | **DOCX**)
- **Think Spot Journal** (**PDF** | **DOCX**)

For the Students

- **Unspotted Bugs** - Online Story
- **First Computer Bug** - Student Video

Vocabulary

- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in your algorithm or program.
- **Persistence** - Trying again and again, even when something is very hard.

Teaching Guide

Warm Up (12 min)

Goal: Help students understand the steps involved in debugging.

Unspotted Bugs

This story can be presented in several ways, including:

- Circled up story time
- Projected with document camera / smartboard
- Pair shared with students at their computers

The story of Unspotted Bugs presents many of the ideas that students will need to understand the debugging process of coding. This warm-up is meant to tie a memorable story together with a concept that young kids often find to be difficult.

Read the book and discuss the techniques that JD used to discover and take care of bugs. Make sure those questions and tactics get repeated often enough that students can recall (if not recite) them without the story in hand.

Lesson Tip

Important ideas from the story:

- What happened?
- What was supposed to happen?
- What does that tell you?
- Did it work at the first step?
- Did it work at the second step?
- Where did it go wrong?

Potential Questions for Storytime:

- Page 3: What do you notice in the picture? What's wrong with the flower? (It's upside down!) What's wrong with the clock? (The hands aren't in the center) Why do you think there is something wrong with these items?(Because there are bugs on them!)
- Page 7: What's wrong with the picture? (The lamp is upside down) Why is that? (There's a bug)
- Page 11: What's wrong in this scene? (The car doesn't have wheels!) Why? (Because there are bugs on it!)
- What did JD find when he went looking for the bug? What was wrong? What does this mean? (JD found an upside down tree. This is wrong because the tree trunk should be touching the ground! This means there is a bug on the tree!)

Vocabulary

This lesson has three new and important vocabulary words:

- *Bug* - Say it with me - Buhh-g. Something that is going wrong. An error.
- *Debugging* - Say it with me: Dee-bug-ing. To find and fix errors.
- *Persistence* - Say it with me: Purr-siss-tense. Not giving up. Persistence works best when you try things many different ways, many different times.

Marble Run Breakdown (10 - 20 min)

Goal: Help students think critically about the difference between what is happening and what is expected.

Debug the Run

Now that students have been introduced to the idea of looking for problems, they can try to apply it to more places in the real world. This next activity gives them practice looking for bugs in Marble Runs (a project that they will be working with next week.)

Grab your sample marble run (built from our plans, or something similar.) Show the students how each piece works, then demonstrate putting them together (but put them together incorrectly, to prevent the ball from flowing properly from A to B.

The goal of this exercise is to help the students identify when something goes wrong, so if they don't catch it the first time, run it again, and again. It can help to make exaggerated frustration faces when the ball doesn't do what you would like it to do.

Let the students share hypotheses about what is going wrong, and how to fix it. Students should feel free to try things that you know will be incorrect. If students misidentify solutions, use the bug finding formula on their configurations. Repeat until you get a working run.

Encouragement is key here. If things don't work right away, praise the class for being so persistent and choosing not to give up. If they start to get frustrated, encourage them to persist a bit longer, promising them that they will get it soon if they just hang in there.

Lesson Tip

Say:

Great! You all are so good at this, maybe you can help me with my own problem!

See, I have this marble run that I made. It comes in two pieces. When I put the ball in here (input A) it's supposed to come out here (output A). When I put the ball in here (input B) it's supposed to come out here (output B). Now, when I slide them together, I should be able to put the ball in here (input A) and have it come out here (output B). But it doesn't work, watch.

[Slide the pieces together with output B facing output A.]

Watch what happens. [Drop ball at input A and notice that it does not come out output B.]

- BUG!

What happened?

- The ball fell on the table.

What was supposed to happen?

- The ball was supposed to drop from A into B.

What does that tell you?

- You should turn B around so that the ball goes into the right place!

Wrap Up (10 - 20 min)

Journaling

Goal: Students will start to understand the importance of the activity they just completed by reflecting on it verbally, then through drawing in their journals.

Clear your mind:

It can be distracting to a learner when they have unanswered questions or doubts. To end this lesson, we'll give everyone the chance to get those out so that they can reflect on what they've been taught.

Encourage students to share their thoughts and questions either with the whole class or with an elbow partner.

Journal Prompts:

Once they've had time to ponder their own thoughts, get the students thinking about the purpose of the lesson that they just learned. Why did you do this activity? How will it help them later? Can they think of buggy things that they've seen in the real world?

Students should finish by drawing or writing in their journal. Possible topics include:

- How do you feel when something that you are working on acts buggy?
- How many times do you think you should try to fix a bug before you give up?
- What would you do if you notice that something is buggy, but you don't know how to fix it?

Lesson Tip

Say:

What do you think we learned in this lesson?

- Debugging
- How to solve a problem
- How to make a marble go
- How do you think that can help us in other places?

Extended Learning

Real Life Bug Hunting

Take your students outside. Do you see any signs of bugs? What are they? Now look closer... can you find the actual bug?

Lesson Tip:

The signs of real-live bugs won't be as dramatic as upside down trees, but it might be dead leaves, spots on flowers, or slime on the
sidewalk. Have the students brainstorm these before going outside to look for them.

Marble Run

Teacher Prep Guide

This guide will provide assistance through a set of two lessons using a Marble Run contraption.

The first portion of this kindergarten series is the debugging lesson, where students will help you debug your Marble Run. In order to do this, you need to have a broken prototype that can be fixed in a predictable way. This guide will suggest an easy step-by-step solution, and give you tips for making a creation using your own design.

For the second half, we are going to ask students to do something incredibly challenging in order to stretch their understanding and aptitude for persistence. This guide will provide additional suggestions and resources to keep the project grade appropriate.

Stage 1: Debugging

The Rules:

The rules of the student version of the Marble Run activity are pretty simple:

1) Build two Marble Runs.
2) Each Marble Run should have at least 3 pieces.
3) Marble Run 1 should take a marble at **Start** height and finish at **Middle** height.
4) Marble Run 2 should take a marble at **Middle** height and finish at **End** height.
5) Put the two Marble Runs together and watch the marble go from **Start** to **End** .

There are a couple of additional rules to adapt this to be effective for the lesson on debugging:

1) The teacher's contraption must not work to begin with.
2) The fix for the issue should be detectable by following the marble's path and determining where the change from "expected" to "unexpected" occurs.

The Set-Up:

Use the Marble Run Ruler (provided on page 2) to determine the starting and ending height for each of the two components, we will call those Component A and Component B.

Component A needs to take in a marble (Input A) at a height that falls somewhere within the highlighted "Start" region. It should then return the marble (Output A) at a height somewhere within the highlighted "Middle" region.

Component B should take a marble (input B) at a height that falls somewhere within the highlighted "Middle" region. It should then return the marble (Output B) at a height somewhere within the highlighted "End" region.

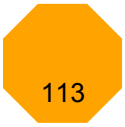
Two simple ways for a teacher to initiate an easy-to-fix failure would be:

A) Have two working components, but connect them in an incorrect way
B) Have Component A release the marble lower than Component B can receive it

Proceed to the teacher guide for Stage 2 for more information on building a Marble Run that falls into either of those categories.

Start  Start

Middle  Middle

End  End

Stage 2: Building a Marble Run

The Rules:

These are the rules for the student version of the Marble Run activity:

1) Build two Marble Runs.
2) Each Marble Run should have at least 3 pieces.
3) Marble Run 1 should take a marble at **Start** height and finish at **Middle** height.
4) Marble Run 2 should take a marble at **Middle** height and finish at **End** height.
5) Put the two Marble Runs together and watch the marble go from **Start** to **End**.

Feel free to change the parameters of these heights as you see fit.

The Set-Up:

Set-up of the student resource area is crucial. Supplies should be plentiful and easy to locate. In addition to the classroom norms (cardstock, tape, safety scissors) volunteers can also donate extra items if given enough notice (paper cups, cereal boxes, and the like).

For further support, place a stack of copies of "Marble Run Hints" (pages 7 & 8) for students to find. You do not need to let the class know that those are available. Students will feel more like they have "discovered" something if the teacher is not involved in the process.

The Build:

We have provided tutorials on four relatively simple pieces that are quite helpful for this project. These peices are:

- **Tube (fig. 1)** - A piece of paper that has been rolled into a cylinder
- **Ramp (fig. 2)** - Paper folded in a zig-zag fashion to provide a ramp with attaching flaps
- **Bridge (fig. 3)** - Paper where two sides have been folded into the center to create a bridge
- **Cone (fig. 4)** - Paper rolled first into a cylinder, then tightened at the bottom and loosened at the top. Once the basic cone has been created, secure with tape, then cut the top and bottom to customize.

A low-frills example contraption can be created using the steps that follow.

Component A:

1) Cut an 8.5"x11" sheet of cardstock in half lengthwise, then cut one of those halves lengthwise again. Fold both of the quarter strips bridge style.
2) Lay the bridges on their sides and tape free edges together to form a square or rectangle.
3) Cut an 8.5"x11" sheet of cardstock in quarters (length, then width). Roll two of the pieces along the long edge and two along the short edge, then secure with tape, to make a total of 4 tubes.
4) Tape the long tubes to the back of the square case from step 2, and the short tubes should be taped in front.

Marble Run

Teacher Prep Guide

Component A (continued):

5) Cut an 8.5”x11” sheet of cardstock in half widthwise. Fold one piece in half lengthwise, then fold the long edges back out toward the crease to make a ramp.
6) Tape the edges of the ramp to the tops of the posts. This gives your main marble path, but it’s not quite tall enough.
7) Add a cone at the intake point, and you’ll be set!

Component B:

8) Cut an 8.5”x11” sheet of cardstock in half widthwise and roll one of the pieces along the short side, to make an 8.5” tube. Secure with tape.
9) Cut the tube at a point anywhere from 2” - 4” at about a 45 degree angle.
10) Rotate one of the pieces to form an elbow, and tape back together.
11) Cut a 1“ strip from the remaining 8.5” x 5.5” cardstock half (lengthwise) and make a bridge to use as a triangular base for the tube to sit in.
12) Use the remaining cardstock to make an input cone for the top of Component B. Trim tube and cone to get appropriate height.

Voila! Your very own Marble Run!

Note: It is highly unlikely that your students will come up with anything this clean and stable. That is OKAY! The version here is meant to be messed with and reused.

Tube

Figure 1

Ramp

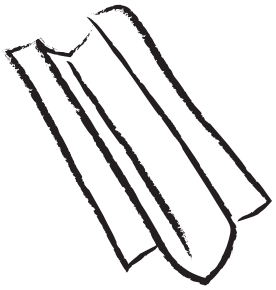


Figure 2

Bridge

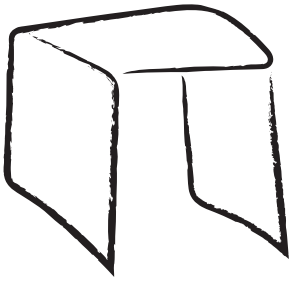


Figure 3

Cone

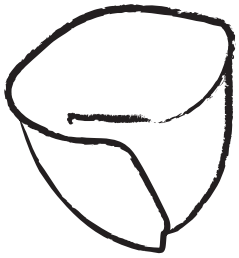


Figure 4

Lesson 2: Persistence & Frustration: Stevie and the Big Project

Unplugged | Fail | Frustrated | Persistence

Overview

When students run into a barrier while answering a question or working on a project, it's so easy for them to get frustrated and give up. This lesson will introduce students to the idea that frustration can be an important part of learning. Here, frustration is presented as a step in the creative process, rather than a sign of failure.

This lession can be done over one or two class sessions. If you have more time, feel free to draw out the building and revising phase of the Marble Run activity.

Purpose

The goal of this lesson is to help students realize that failure and frustration are common when working on projects, but that doesn't mean that they should give up.

In this lesson, students will develop an understanding of what it means to be frustrated while working on a large project. It's possible that not every student will experience frustration with this activity, but there are many opportunities to open a discussion about moments in the past where students have felt frustrated but nevertheless persisted.

Agenda

Warm Up (15 min)

- **Stevie and the Big Project**
- **Vocabulary**

Marble Run (20 - 45 min)

- **Before the Project:**
- **Building the Marble Run:**
- **After the Marble Run:**

Wrap Up (5 min)

- **Journaling**

Extended Learning

Objectives

Students will be able to:

- Recognize and point out symptoms of frustration.
- Describe at least one reason why they will choose to be persistent in the face of frustration, rather than giving up.

Preparation

- [ ] Pre-read "Stevie and the Big Project" to identify appropriate questions for your class.
- [ ] Follow instructions in the **Marble Run - Teacher Prep Guide** to make a Marble Run.
- [ ] Print copies of the **Marble Run Ruler** (page 2 of teacher guide) for each student or pair of students
- [ ] Prepare a resource station with cardstock, safety scissors, tape, and anything else you think might be fun for students to build with. Include a stack of the **"Marble Run Hints"** pages from the Teacher Prep Guide, but do not advertise their existence.
- [ ] (Optional) Allow students to bring cardboard, popsicle sticks, string, or other tidbits from home to add to the resource station.
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **Marble Run** - Teacher Prep Guide (**PDF** | **DOCX**)
- **Think Spot Journal** (**PDF** | **DOCX**)

For the Students

- **Stevie and the Big Project** - Online Story

Vocabulary

- **F.A.I.L.** - First Attempt In Learning
- **Frustrated** - Feeling annoyed or angry because something is not the way you want it.
- **Persistence** - Trying again and again, even when something is very hard.

Teaching Guide

Warm Up (15 min)

Stevie and the Big Project

Goal: Introduce students to the idea that they don't have to give up just because they are frustrated.

This lesson begins with a story. Students will be introduced to several ideas on persistence and frustration through relatable struggles by fictional characters, including the idea that frustration is not a sign that someone should instantly give up.

This book can be presented in several ways, including:

- Circled up story time
- Projected with document camera / smartboard
- Pair share with students at their computers

Use the reading techniques that work in your classroom:

If your students like to discuss things that happen as they appear in the book, be sure to stop your class after large plot areas like when Stevie breaks her structure, or when Laurel explains frustration.

If your students like to sit through a whole story and discuss at the end, read through the book, then prompt their memory with some "Remember when…" type questions.

Lesson Tip

Sample Questions:

- How would you feel if you were given a project that feels much harder than what you are used to?
- Do you think it's okay to try something new, even if it doesn't work out the first time?
- Why do you think Stevie smashed her project?
  - Do you think that helped her or hurt her when it comes to reaching her goal?
  - What do you think Stevie should have done instead of breaking her project?
- Can somebody explain what frustration is?
- How do you think you can know when you are frustrated?
  - What face do you make when you are frustrated?
  - How can you make yourself feel better when you start to get frustrated?
  - We all get frustrated sometimes. Does that mean that we should give up?
- Can someone tell me what persistence is?
  - Why is it hard to learn if you're not persistent?
  - Can you tell me why you might be tempted not to be persistent?
  - What happened when Stevie decided to be persistent?
  - Do you think you can be persistent?

Vocabulary

- *Persistence* - Say it with me: Purr-siss-tense. Not giving up. Persistence works best when you try things many different ways, many different times.
- *Frustrated* - Say it with me: Frus - straight - ted. Feeling annoyed or angry because something is not the way you want it.
- *F.A.I.L.* - First Attempt in learning. When you try to do something, but you don't do it quite right.

Marble Run (20 - 45 min)

This activity is meant to highlight and normalize the feeling of frustration, while giving students a chance to be persistent.

Before the Project:

It is vitally important that students understand that this activity is meant to help them learn about frustration and persistence. This is not one of those times when we allow students to experience something, then give it a name afterward. Students need to know that they will be feeling some emotions, and that those emotions are okay.

Take a moment to relate the next activity back to the book that you just read. The class might be excited that they get to try the same project that Stevie did, but they might also be apprehensive at the thought of tackling something difficult.

Encourage your students to have their Think Spot Journals around during the activity so they can use them to plan, solve, and voice concerns.

Building the Marble Run:

Time to be an engineer!

Break students up into pairs and have them quickly come up with a team name. This should help to unify them in their work.

Next, point out the resource station that you have set up with all of the supplies and goodies that students will have access to. Make sure you are very clear about whether they are limited only to the items in the resource station or whether they are allowed to ask for other items for their creation.

Give students checkpoints for this activity. Make sure that they know that there is no penalty for not finishing on time.

Preplanning is optional, since prediction is not often a kindergartener's strong suit.

The first attempt at building will likely be hectic and a bit sloppy, but it should give students access to the feelings and opportunities for persistence that are being studied in this lesson.

Try to end the Marble Run build with an opportunity for groups to collaborate. This will improve the chances of success for students who have been struggling, without the need for teacher intervention.

After the Marble Run:

Time to do some damage control if any is needed.

Remind students that this activity was planned to teach students how to identify feelings of frustration and work past them to be persistent.

Discuss the difference between being successful for the purpose of this activity, and being successful at building their contraption. Is it possible to have done the first without the second?

Lesson Tip

Say:

Now, we're going to do something very fun, and very challenging! I am going to let you all try to make a Marble Run of your own!

This is **supposed** to be challenging. That's part of the fun! Your Marble Run probably won't work right the first time, and that's alright. The goal for this game is to practice being persistent.

Remember, Stevie showed us that this might be difficult, and sometimes difficult things are frustrating. It is okay if you get frustrated during this activity. Most of us probably will at some point. How should we handle those feelings?

- Count to 10
- Take deep breaths
- Journal about them
- Talk to a partner about them
- Ask for help

Lesson Tip

Checkpoint Suggestions:

- Pre-planning time (3-5 minutes)
- First attempt at building (10-15 minutes) -- For a longer (or two day) time period --
- Discuss with another group (3-5 minutes)
- Revision of structure (10-15 minutes) -- Wrap Up Work --
- Collaborative work time (5-15 minutes)

Teacher Tip

Tears are a very common byproduct when kindergarteners attempt lessons of this nature. You will likely want to have a pre-packaged prescription for students who become emotionally raw.

- Can you put into words what you are feeling right now?
- Stevie would be so proud of you. What do you think Laurel and Jorge would say if you told them how you feel?
- What would it be called if you said out loud that you are frustrated, but decided to keep working anyway?
  - Do you feel like you can be persistent with me today?

Wrap Up (5 min)

Journaling

Allow students to reflect on the emotions and processes experienced during the lesson.

Finish out this lesson by asking students to spend some time in their **Think Spot Journal**.

Journal Prompts:

- Draw a picture of what you look like when you're frustrated.
- Draw a picture that shows things you can do to feel better when you're frustrated.
- What does persistence look like?

Extended Learning

- Add a third piece to the beginning of the Marble Run. Can students start a marble up even higher and get it to flow through the rest of their contraption?

- Talking through frustration. Can students think of things that they can say to classmates to help them be persistent when they are frustrated?

Marble Run

Teacher Prep Guide

This guide will provide assistance through a set of two lessons using a Marble Run contraption.

The first portion of this kindergarten series is the debugging lesson, where students will help you debug your Marble Run. In order to do this, you need to have a broken prototype that can be fixed in a predictable way. This guide will suggest an easy step-by-step solution, and give you tips for making a creation using your own design.

For the second half, we are going to ask students to do something incredibly challenging in order to stretch their understanding and aptitude for persistence. This guide will provide additional suggestions and resources to keep the project grade appropriate.

Stage 1: Debugging

The Rules:

The rules of the student version of the Marble Run activity are pretty simple:

1) Build two Marble Runs.
2) Each Marble Run should have at least 3 pieces.
3) Marble Run 1 should take a marble at **Start** height and finish at **Middle** height.
4) Marble Run 2 should take a marble at **Middle** height and finish at **End** height.
5) Put the two Marble Runs together and watch the marble go from **Start** to **End** .

There are a couple of additional rules to adapt this to be effective for the lesson on debugging:

1) The teacher's contraption must not work to begin with.
2) The fix for the issue should be detectable by following the marble's path and determining where the change from "expected" to "unexpected" occurs.

The Set-Up:

Use the Marble Run Ruler (provided on page 2) to determine the starting and ending height for each of the two components, we will call those Component A and Component B.

Component A needs to take in a marble (Input A) at a height that falls somewhere within the highlighted "Start" region. It should then return the marble (Output A) at a height somewhere within the highlighted "Middle" region.

Component B should take a marble (input B) at a height that falls somewhere within the highlighted "Middle" region. It should then return the marble (Output B) at a height somewhere within the highlighted "End" region.

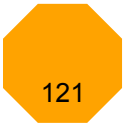
Two simple ways for a teacher to initiate an easy-to-fix failure would be:

A) Have two working components, but connect them in an incorrect way
B) Have Component A release the marble lower than Component B can receive it

Proceed to the teacher guide for Stage 2 for more information on building a Marble Run that falls into either of those categories.

Start  Start

Middle  Middle

End  End

Stage 2: Building a Marble Run

The Rules:

These are the rules for the student version of the Marble Run activity:

1) Build two Marble Runs.
2) Each Marble Run should have at least 3 pieces.
3) Marble Run 1 should take a marble at **Start** height and finish at **Middle** height.
4) Marble Run 2 should take a marble at **Middle** height and finish at **End** height.
5) Put the two Marble Runs together and watch the marble go from **Start** to **End**.

Feel free to change the parameters of these heights as you see fit.

The Set-Up:

Set-up of the student resource area is crucial. Supplies should be plentiful and easy to locate. In addition to the classroom norms (cardstock, tape, safety scissors) volunteers can also donate extra items if given enough notice (paper cups, cereal boxes, and the like).

For further support, place a stack of copies of "Marble Run Hints" (pages 7 & 8) for students to find. You do not need to let the class know that those are available. Students will feel more like they have "discovered" something if the teacher is not involved in the process.

The Build:

We have provided tutorials on four relatively simple pieces that are quite helpful for this project. These peices are:

- **Tube (fig. 1)** - A piece of paper that has been rolled into a cylinder
- **Ramp (fig. 2)** - Paper folded in a zig-zag fashion to provide a ramp with attaching flaps
- **Bridge (fig. 3)** - Paper where two sides have been folded into the center to create a bridge
- **Cone (fig. 4)** - Paper rolled first into a cylinder, then tightened at the bottom and loosened at the top. Once the basic cone has been created, secure with tape, then cut the top and bottom to customize.

A low-frills example contraption can be created using the steps that follow.

Component A:

1) Cut an 8.5"x11" sheet of cardstock in half lengthwise, then cut one of those halves lengthwise again. Fold both of the quarter strips bridge style.
2) Lay the bridges on their sides and tape free edges together to form a square or rectangle.
3) Cut an 8.5"x11" sheet of cardstock in quarters (length, then width). Roll two of the pieces along the long edge and two along the short edge, then secure with tape, to make a total of 4 tubes.
4) Tape the long tubes to the back of the square case from step 2, and the short tubes should be taped in front.

Marble Run

Teacher Prep Guide

Component A (continued):

5) Cut an 8.5”x11” sheet of cardstock in half widthwise. Fold one piece in half lengthwise, then fold the long edges back out toward the crease to make a ramp.
6) Tape the edges of the ramp to the tops of the posts. This gives your main marble path, but it’s not quite tall enough.
7) Add a cone at the intake point, and you’ll be set!

Component B:

8) Cut an 8.5”x11” sheet of cardstock in half widthwise and roll one of the pieces along the short side, to make an 8.5” tube. Secure with tape.
9) Cut the tube at a point anywhere from 2” - 4” at about a 45 degree angle.
10) Rotate one of the pieces to form an elbow, and tape back together.
11) Cut a 1“ strip from the remaining 8.5” x 5.5” cardstock half (lengthwise) and make a bridge to use as a triangular base for the tube to sit in.
12) Use the remaining cardstock to make an input cone for the top of Component B. Trim tube and cone to get appropriate height.

Voila! Your very own Marble Run!

Note: It is highly unlikely that your students will come up with anything this clean and stable. That is OKAY! The version here is meant to be messed with and reused.

Tube

Figure 1

Ramp

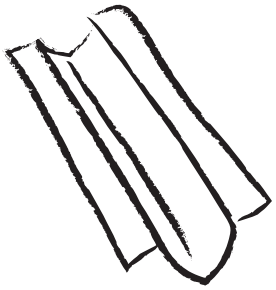


Figure 2

Bridge

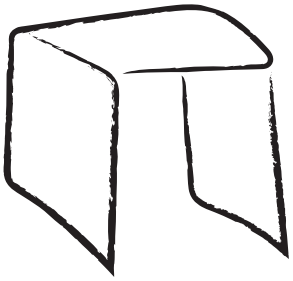


Figure 3

Cone

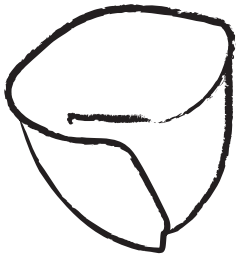
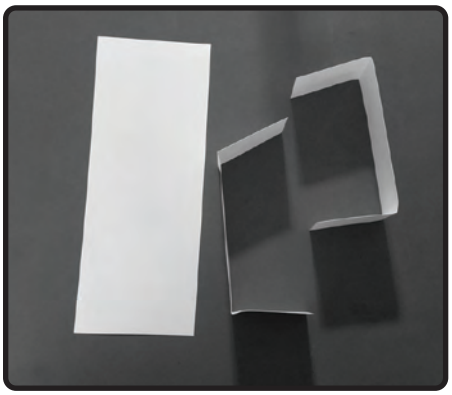


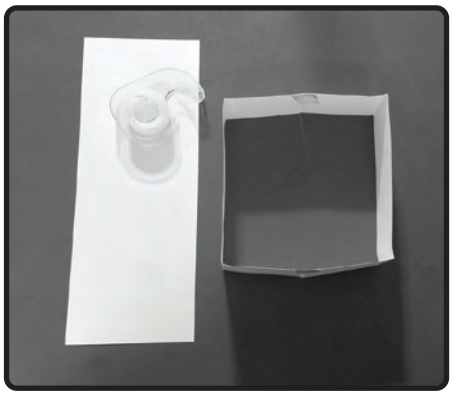
Figure 4

Marble Run

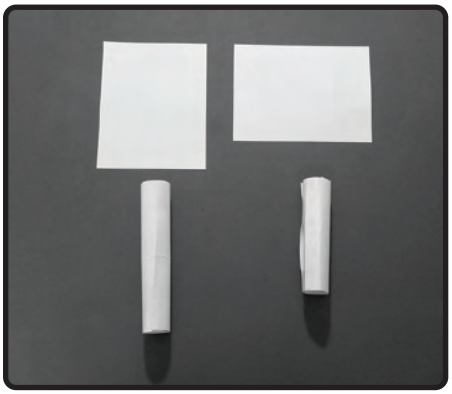
Teacher Prep Guide



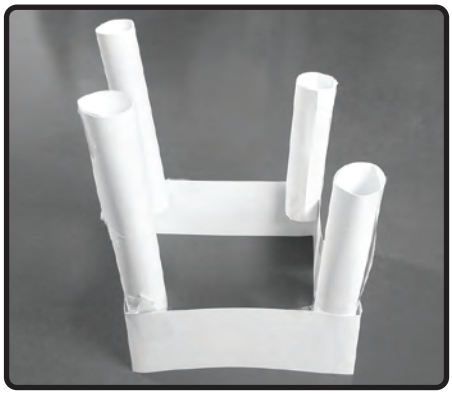
Step 1: Fold strips "bridge style"



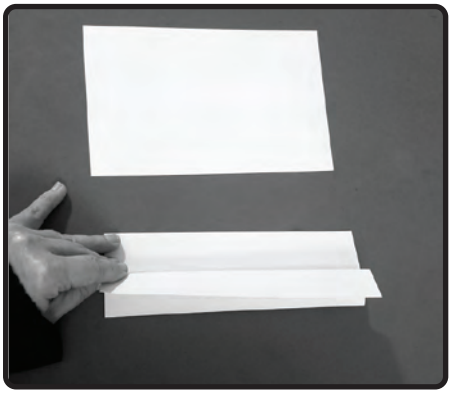
Step 2: Tape ends of folded strips together to make a base.



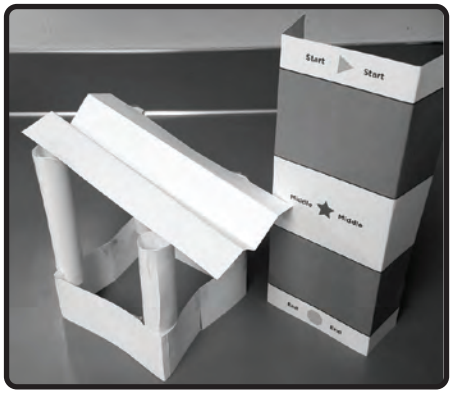
Step 3: Roll quartered paper into tubes and secure with tape.



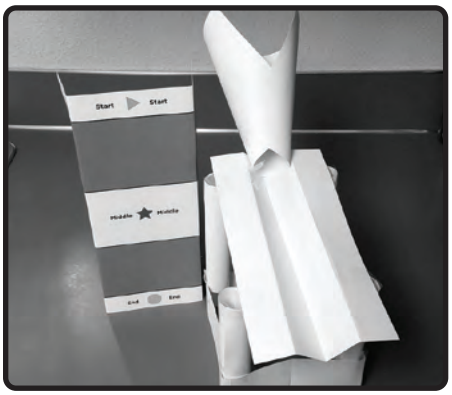
Step 4: Tape tubes inside case. Make sure to tape them near the top for height.



Step 5: Make a ramp from a half sheet of cardstock.

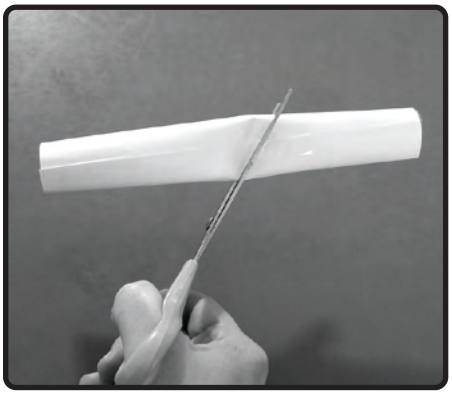


Step 6: Tape ramp to base and check height.

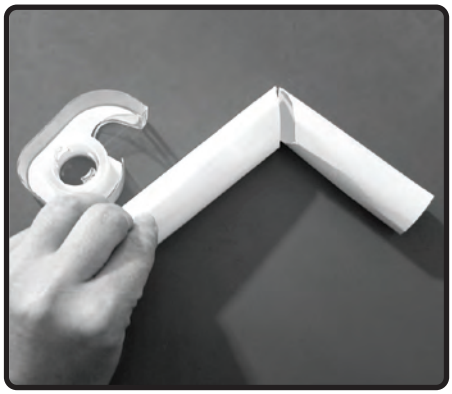


Step 7: Add cone to finish Component A.

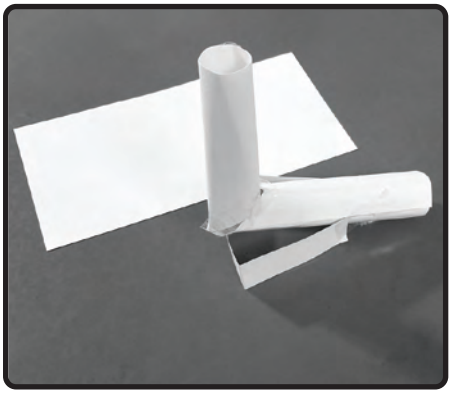
Step 8: Roll 1/2 sheet into tube to start on Component B.



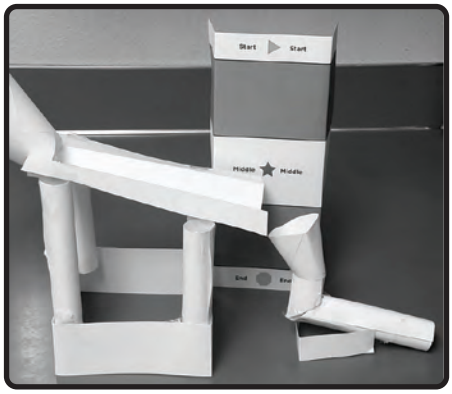
Step 9: Cut tube at an angle.



Step 10: Tape tubes back together to make an elbow.



Step 11: Make a base from a thin strip of cardstock.



Step 12: Add a cone to the top and trim pieces to size.

H

Handout

Marble Run Hints

Student Handout

C O D E

Try using some of these:

Tube

Ramp

Bridge

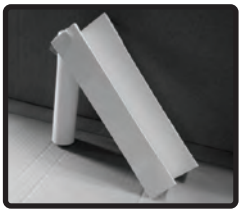
Cone

Marble Run Hints

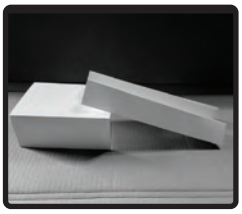
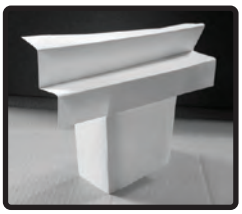
(Continued)

Now try putting them together!

Tube + Ramp →

 or 

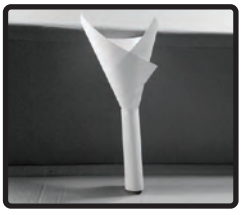
Ramp + Bridge →

 or 

Bridge + Tube →

 or 

Cone + Tube →

 or 

Lesson 3: Real-Life Algorithms: Plant a Seed

Unplugged | Algorithms

Overview

In this lesson, students will relate the concept of algorithms back to everyday, real-life activities by planting an actual seed. The goal here is to start building the skills to translate real-world situations to online scenarios and vice versa.

Purpose

In this lesson, students will learn that algorithms are everywhere in our daily lives. For example, it is possible to write an algorithm to plant a seed. Instead of giving vague or over-generalized instructions, students will break down a large activity into smaller and more specific commands. From these commands, students must determine a special sequence of instructions that will allow their classmate to plant a seed.

Agenda

Warm Up (10 min)

- **Vocabulary**
- **What We Do Daily**

Main Activity (20 min)

- **Real-Life Algorithms: Plant A Seed - Worksheet**

Wrap Up (10 - 20 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (15 min)

- **Real-Life Algorithms - Assessment**

Extended Learning

- **Go Figure**

Objectives

Students will be able to:

- Decompose large activities into a series of smaller events.
- Arrange sequential events into their logical order.

Preparation

- [ ] Watch the **Plant a Seed - Teacher Video**.
- [ ] Prepare supplies for planting seeds. You'll need seeds, dirt, and paper cups for each student or group.
- [ ] Print one **Real-Life Algorithms: Plant A Seed - Worksheet** for each student.
- [ ] Print one **Real-Life Algorithms - Assessment** for each student.
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **Real-Life Algorithms: Plant A Seed** - Worksheet
- **Real-Life Algorithms** - Assessment
- **Plant a Seed** - Teacher Video

Teaching Guide

Warm Up (10 min)

Vocabulary

This lesson has one vocabulary word that is important to review:

Algorithm - Say it with me: Al-go-ri-thm

A list of steps that you can follow to finish a task

What We Do Daily

- Ask your students what they did to get ready for school this morning.
  - Write their answers on the board
  - If possible, put numbers next to their responses to indicate the order that they happen
    - If students give responses out of order, have them help you put them in some kind of logical order
    - Point out places where order matters and places where it doesn't
- Introduce students to the idea that it is possible to create algorithms for the things that we do everyday.
  - Give them a couple of examples, such as making breakfast, tying shoes, and brushing teeth.
- Let's try doing this with a new and fun activity, like planting a seed!

Main Activity (20 min)

Real-Life Algorithms: Plant A Seed - Worksheet

You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to help each other plant a seed. Directions:

- Cut out the steps for planting a seed from the **Real-Life Algorithms: Plant A Seed - Worksheet**.
- Work together to choose the six correct steps from the nine total options.
- Glue the six correct steps, in order, onto a separate piece of paper.
- Trade the finished algorithm with another person or group and let them use it to plant their seed!

Lesson Tip

You know your classroom best. As the teacher, decide if you should all do this together, or if students should work in pairs or small groups.

Wrap Up (10 - 20 min)

Flash Chat: What did we learn?

- How many of you were able to follow your classmates' algorithms to plant your seeds?
- Did the exercise leave anything out?
  - What would you have added to make the algorithm even better?
  - What if the algorithm had been only one step: "Plant the seed"?
    - Would it have been easier or harder?
    - What if it were forty steps?
- What was your favorite part about that activity?

Lesson Tip

If deciding on the correct steps seems too difficult for your students, do that piece together as a class before you break up into teams.

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- Draw the seed you planted today.
- Write the algorithm you used to plant the seed.

Assessment (15 min)

Real-Life Algorithms - Assessment

- Hand out the worksheet titled **Real-Life Algorithms - Assessment** and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Go Figure

- Break the class up into teams.
- Have each team come up with several steps that they can think of to complete a task.
- Gather teams back together into one big group and have one team share their steps, without letting anyone know what the activity was that they had chosen.
- Allow the rest of the class to try to guess what activity the algorithm is for.

Name: _____ Date: _____

Real-Life Algorithms

Plant a Seed Worksheet

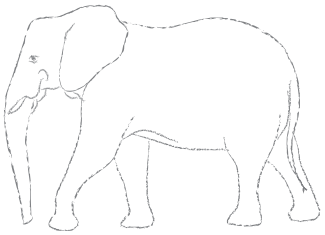
Unplugged

You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to help each other plant a seed.

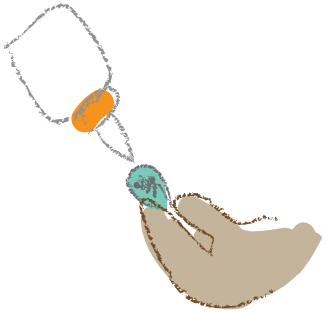
Cut out the steps of planting a seed below, then work together to glue the six the correct steps, in order, onto a separate piece of paper. Trade your finished algorithm with another person or group and let them use it to plant their seed!

PUT POT IN SUNLIGHT

PUT SEED IN HOLE



HUG AN ELEPHANT



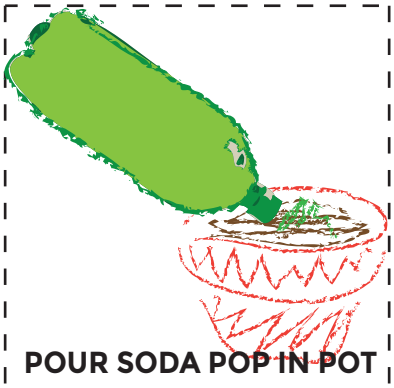
PUT GLUE ON SEED

FILL POT WITH SOIL

POKE HOLE IN SOIL

WATER POT

COVER SEED WITH SOIL



POUR SODA POP IN POT

U

Unplugged

Name: _____ Date: _____

Real-Life Algorithms

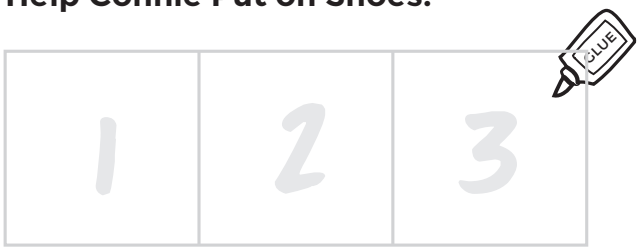
Assessment Worksheet

C O D E

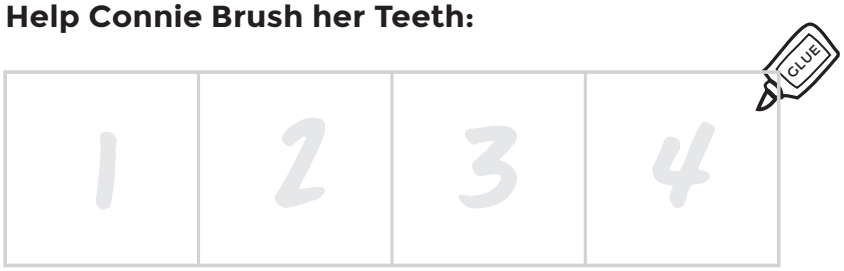
An algorithm is a list of steps that you can follow to finish a task. We follow algorithms every day when it comes to activities like making the bed, making breakfast, or even getting dressed in the morning.

Connie the Coder just woke up and is still feeling very sleepy. Can you put together some algorithms to help Connie get ready for the day?

Help Connie Put on Shoes:



Help Connie Brush her Teeth:



Help Connie Plant a Seed:

Name: _____ Date: _____

Real-Life Algorithms

Assessment Worksheet

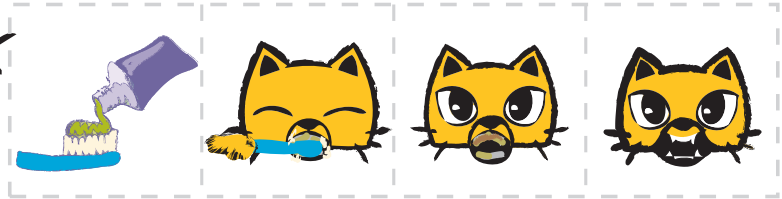
C O
D E

These items are out of order. To help Connie, cut out each picture and rearrange them into the right sequence in the category above.

Putting on Shoes

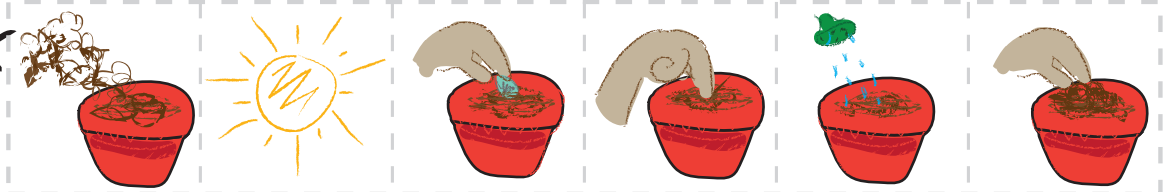
PUT FOOT IN SHOE | PICK UP SHOES | TIE SHOE

Brusing Teeth



PASTE ON BRUSH | BRUSH TEETH | DIRTY TEETH | CLEAN TEETH

Plant a Seed



FILL POT WITH SOIL | PUT POT IN SUNLIGHT | PUT SEED IN HOLE | POKE HOLE IN SOIL | WATER POT | COVER SEED WITH SOIL

Lesson 5: Common Sense Education: Your Digital Footprint

Common Sense Education | Unplugged

Overview

In collaboration with **Common Sense Education - Website**, this lesson helps students learn about the similarities of staying safe in the real world and when visiting websites. Students will also learn that the information they put online leaves a digital footprint or "trail." This trail can be big or small, helpful or hurtful, depending on how they manage it.

Purpose

Common Sense Education has created this lesson to teach kids the importance of understanding the permanence of something posted on the internet. By relating footprints on a map to what a student might post online, students will make important connections between being tracked by a physical footprint on a path and being tracked based on information posted online.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Pause and Think**

Main Activity (20 min)

- **Follow the Digital Trail - Worksheet**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (5 min)

- **Digital Footprint - Assessment**

Extended Learning

Objectives

Students will be able to:

- Understand that being safe when they visit websites is similar to staying safe in real life.
- Learn to recognize websites that are safe for them to visit.
- Recognize if they should ask an adult they trust before they visit a particular website.
- Explore what information is appropriate to be put online.

Preparation

- [ ] Watch this **Your Digital Footprint - Teacher Video**.
- [ ] Prepare to show **Your Digital Footprint - Lesson Video**.
- [ ] (Optional) Prepare to show **Pause and Think Online - Video**.
- [ ] Common Sense Education's **Follow the Digital Trail - Worksheet** game.
- [ ] Print one **Animal Tracks** chart (page 7) for each student.
- [ ] Print one **Digital Footprint - Assessment** for each student.

Links

For the Teacher

- **Your Digital Footprint** - Teacher Video
- **Your Digital Footprint** - Lesson Video
- **Follow the Digital Trail** - Worksheet
- **Digital Footprint** - Assessment
- **Common Sense Education** - Website
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Digital Footprint** - The collected information about an individual across multiple websites on the Internet.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important phrase:

- **Digital Footprint** - Say it with me: Dih-jih-tal Foot-print

The information about someone on the internet.

Pause and Think

- Ask What does it mean to be safe?
- When you walk down the street or play in your neighborhood without a trusted adult there, how do you stay safe?
- Tell students that just as they should stay safe in the real world, they should stay safe when they go into the online world (visiting websites). Make parallels between the answers students gave you about their neighborhood and the online world.

Play the **Your Digital Footprint - Lesson Video**.

- Introduce the idea that there are three different kinds of websites that students may have the opportunity to visit.
  - Green: A “green” website is:
    - A good site for kids your age to visit
    - Fun, with things for you to do and see
    - Has appropriate words
    - Doesn’t let you talk to people you don’t know
  - Yellow: A “yellow” website is:
    - A site you are not sure is right for you
    - One that asks for information such as who you are, where you live, your phone number or email address, etc.
    - A place where you are allowed to communicate freely with others
  - Red: A “red” website is:
    - A site that is not right for you
    - A place you might have gone to by accident
    - Filled with things that are for older kids or adults
  - Discuss examples of each of these kinds of sites.

Lesson Tip

If you have access to a computer, feel free to navigate to sites that might showcase each of these types (using extreme caution with your RED selection).

Now, let's see what we can do to keep ourselves safe.

Main Activity (20 min)

common sense education®

Follow the Digital Trail - Worksheet

- Peruse the **Follow the Digital Trail - Worksheet** lesson on the Common Sense Education webpage.
- Give each student an **Animal Tracks Chart** (page 7).

	Mizzle the Mouse	Electra the Elephant
Whose full name do you know?		
Whose house could you find?		
Whose birth date do you know?		
Whose user name and password do you know?		
Who let out a secret on the internet?		
Which animal can you describe better from his or her photo?		

Directions:

- Place the *Digital Trail Squares* on the ground, face down, in two different trails, keeping Mizzle the Mouse and Electra the Elephant's trails separate from one another.
- Share the stories of Mizzle and Electra. These animals decided it would be fun to put some information about themselves online. They went onto **www.wildkingdom.com** and posted information. The only problem is that they forgot to ask their parents if it was okay first.
- Explain to students that they are from the "Things Big and Small" Detective Agency. A hunter has hired them to find out as much as possible about Mizzle the Mouse and Electra the Elephant. The more the detectives learn, the better for their plan to take over the animal kingdom.
- Divide students into groups of four. Tell them that each group should have a detective that will keep detailed notes.
- Invite students to go on a hunt for information. Let them know that the information that Mizzle and Electra post can be seen by anyone, including the detectives. Each group should follow the digital trail of both animals, starting with the mouse and then the elephant. Stagger the groups so they are on the trail at slightly different times. Students should fill out their handout as they go.

Lesson Tip

If your students have trouble writing, feel free to do this activity as a group and have students raise their hand when they find clues. This will allow you (or a teacher aide) to help communicate and record the information being shared.

For more in-depth modules, you can find additions to this curriculum at the **Common Sense Education - Website** page on Scope and Sequence.

Wrap Up (15 min)

Flash Chat: What did we learn?

- Who can the detectives find out more about, and why?
- Which animal has a bigger digital footprint?
- Mizzle says some interesting things about himself on the Internet. What are they?
- Is there anything that Electra posted on the Internet that could become a problem for her? If so, what and why?

Take the time to discuss what is appropriate information to share on the Internet, and what is not:

Lesson Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow-partner.

Appropriate	Not Appropriate
Interests	Address
Hobbies	Full Name
First Name	Information that would hurt others

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw some things that you should never talk to a stranger about on the internet. For example, draw your house to represent your address, draw your school, or draw your family.

Assessment (5 min)

Digital Footprint - Assessment

Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Common Sense Education

- Visit **Common Sense Education - Website** to learn more about how you can keep your students safe in this digital age.

Follow the Digital Trail

Essential Question

What information is appropriate in a digital footprint?

Lesson Overview

Students learn that the information they put online leaves a digital footprint or "trail." This trail can be big or small, helpful or hurtful, depending on how they manage it.

Students follow the digital information trails of two fictional animals. They make observations about the size and content of each trail, and connect these observations by thinking critically about what kinds of information they want to leave behind.

Learning Objectives

Students will be able to ...

- learn that the information they put online leaves a digital footprint or "trail."
- explore what information is appropriate to be put online.
- judge the nature of different types of digital footprints by following the information trails of two fictional animals.

Materials and Preparation

- Cut apart the Digital Trail Squares (found at the end of the lesson plan), keeping the elephant and mouse squares separate. Be prepared to lay out each animal's "tracks" in different locations in the classroom after the lesson introduction.
- Copy the **Animal Tracks Student Handout**, one for each group of four.

Family Resources

- Send home the **Privacy and Digital Footprints Family Tip Sheet (Elementary School)**.

Estimated time: 45 minutes

Standards Alignment –

Common Core:

grade K: RL.1, RL.3, RL.4, RL.10, RI.1, RI.4, RI.10, RF.4, W.2, W.5, W.7, W.8, W.10, SL.1a, SL.1b, SL.2, SL.3, SL.4, SL.6, L.6

grade 1: RL.1, RL.3, RL.4, RI.1, RI.4, RI.10, RF.4a, W.5, W.7, W.8, L6

grade 2: RL.1, RL.3, RI.4, RI.10, RF.4a, W.2, W.5, W.7, W.8, SL.1a, SL.1b, SL.1c, SL.3, SL.6, L.6

NETS•S: 1a, 1d, 2d, 3d, 4a-c

Key Vocabulary –

trail: a path or track

digital footprint: the informatiom about you on the Internet

permanent: there forever

introduction

Warm-up *(5 minutes)*

SHARE with students that you can place information online much like you pin something to a bulletin board.

ASK:

What kinds of things are on the bulletin board or walls in our classroom?	Sample responses: • Student work • Photos of students • Birthday chart

INVITE students to imagine that all of the information on the walls of their classroom was pinned up on a bulletin board at a local grocery store.

ASK:

Would you be comfortable with this information being up for everyone to see?	Guide students to think about how some information is better kept for only their eyes or the eyes of people close to them.

EXPLAIN that there is certain information that might be fine to show anyone. But there is also personal and private information – such as their addresses, birth dates, and photos of their family vacations – which is not meant for most people's eyes.

teach 1

Follow the Digital Trail *(15 minutes)*

DEFINE the Key Vocabulary term **trail**.

PLACE the Digital Trail Squares on the ground, face down, in two different trails, keeping Mizzle the Mouse's and Electra the Elephant's trails separate from each another.

SHARE the stories of Mizzle and Electra. These animals decided it would be fun to put some information about themselves online. They went onto www.wildkingdom.com and posted information. The only problem is that they forgot to ask their mamas if it was okay first.

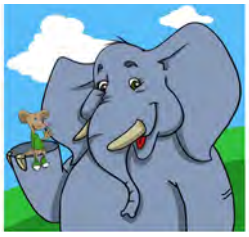
EXPLAIN to students that they are from the Things Big and Small Detective Agency. An evil human has hired them to find out as much as possible about Mizzle the Mouse and Electra the Elephant. The more the detectives learn, the better for their plan to take over the animal kingdom.

DIVIDE students into groups of four. Tell them that each group should have a detective that will keep detailed notes.

DISTRIBUTE the **Animal Tracks Student Handout** to each group.

INVITE students to go on a hunt for information. Let them know that the information that Mizzle and Electra post can be seen by anyone, including the detectives. Each group should follow the digital trail of both animals, starting with the mouse and then the elephant. Stagger the groups so they follow the trails at slightly different times. Students should fill out their handout as they go.

teach 2



Digital Footprints *(20 minutes)*

INVITE each group to report to the rest of the class what they learned about each of the animals, using the **Animal Tracks Student Handout**.

	Mizzle the Mouse	Electra the Elephant
1. Whose full name do you know?		x
2. Whose house could you find?		x
3. Whose birth date do you know?		x
4. Whose username and password do you know?		x
5. Who let out a secret on the Internet?		x
6. Which animal can you describe better from his or her photo?		x

DEFINE the Key Vocabulary terms **digital footprint** and **permanent**.

ASK:

Who can the detectives find out more about, and why?	Electra, because we now know where Electra lives, what she looks like, and private and personal information about her life. Point out to students that having a bigger digital footprint means the detectives can learn more about them too.
Which animal has a bigger digital footprint?	Electra, because she put more private and personal information online than Mizzle.
Mizzle says some funny things about himself on the Internet. What are they?	He says he likes Swiss cheese, his photo is of cheese, and he has a pet flea.
Is there anything that Electra posted on the Internet that could become a problem for her? If so, what and why?	Private and personal information (e.g., address, full name) allows others to learn more about her. This could be unsafe. Saying that she fights with her brother could hurt her brother's feelings because it is public.)

CREATE a chart with students that summarizes which information is okay to share online and what is not okay.

Okay to Share	NOT Okay to Share
Interests	Address
Hobbies	Full name
First name	Information that would hurt others

DISCUSS how Mizzle and Electra both had very interesting information online, but Mizzle used better judgment about what was most appropriate to post. Mizzle had a smaller digital footprint. Electra put some information online that might make her unsafe or might upset her brother.

REMIND students that the Internet is a public space where people they do not know will likely see their information. And this information is very hard to remove. It is basically permanent.

EMPHASIZE that it's important for students to ask their parents or caregives for permission before sharing information about themselves online.

closing

Wrap-up *(5 minutes)*

You can use these questions to assess your students' understanding of the lesson objectives.

ASK:

What is a digital footprint, and what did Mizzle's and Electra's footprints look like?	A digital footprint is the information about you on the Internet. Mizzle's footprint is pretty small and does not reveal private or personal information. Electra's is large and contains information that could make her unsafe or upset others.
What kind of information is okay to share on the Internet? What kind of information is NOT okay to share on the Internet?	*Appropriate*: interests, hobbies, first name. *Inappropriate*: full name, address, hurtful information about others.
Can you put interesting and funny information online and still be appropriate?	Absolutely. Just look at the information that Mizzle posted.

Digital Trail Squares

Mizzle the Mouse

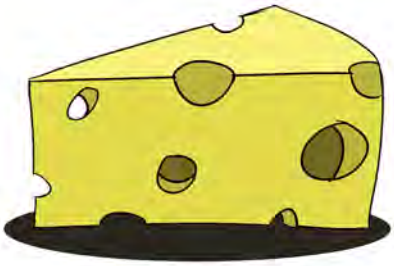
Name:
Mizzle

Where you live:
Mouse hole

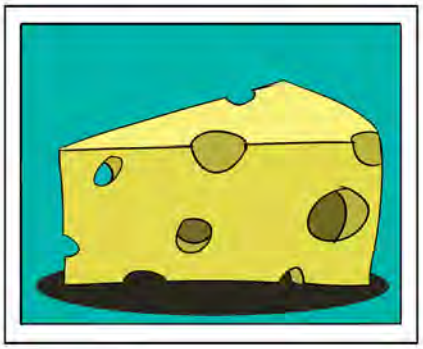
Pet's name:
Frank the Flea

Favorite hobby:
Ice skating

Favorite food:
Cheese



My favorite photo:



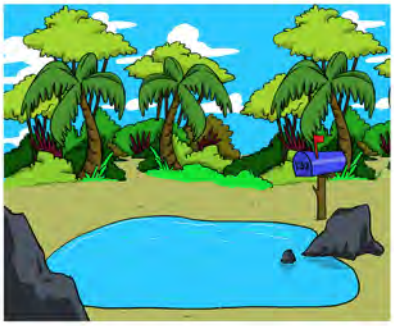
Electra the Elephant

Name:

My full name is:
Electra Ella Elephant

Where you live:

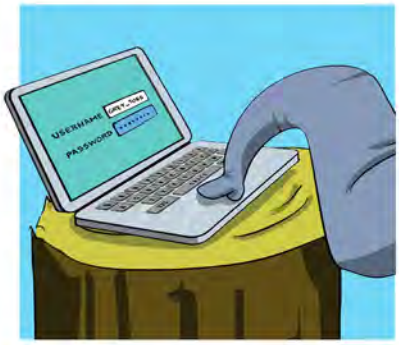
123 Water Hole Lane,
Peanuts, Ohio



Birth date:

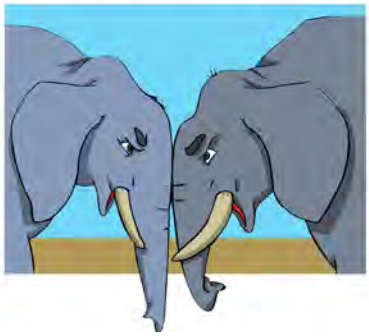
February 21, 2010

User name: gray_toes
Password: bamboo



Secret:

My brother and I
fight all the time



My favorite photo:

Follow The Digital Trail

Directions

Follow the trails of Mizzle the Mouse and Electra the Elephant. Fill in the chart below. Then answer the questions.

	Mizzle the Mouse	Electra the Elephant
1. Whose full name do you know?		
2. Whose house could you find?		
3. Whose birth date do you know?		
4. Whose username and password do you know?		
5. Who let out a secret on the Internet?		
6. Which animal can you describe better from his or her photo?		

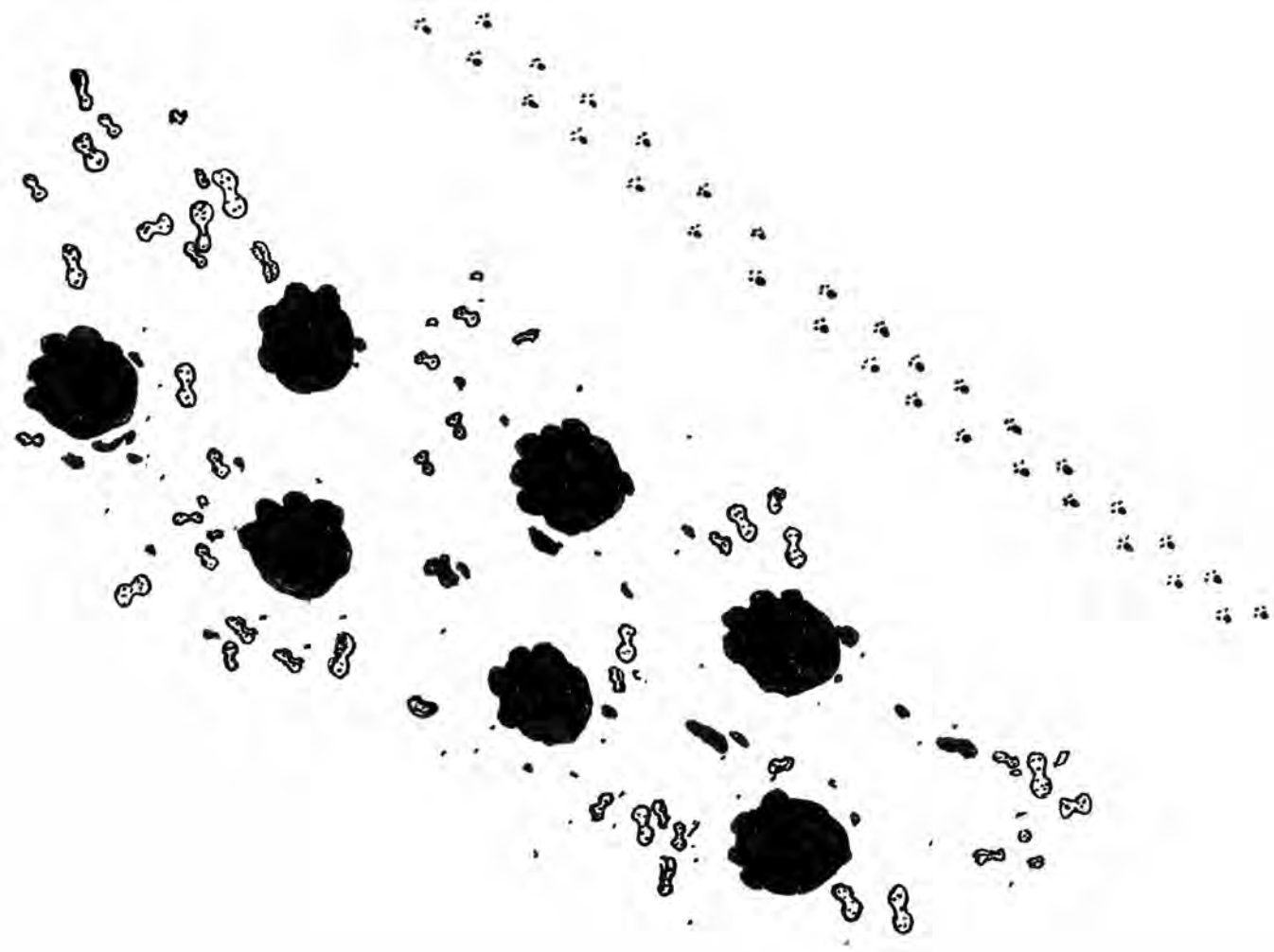
Question

1. Who can the detectives find out more about, and why?

2. Which animal has a bigger digital footprint?

3. Mizzle says some funny things about himself on the Internet. What are they?

4. Is there anything that Electra posted on the Internet that could become a problem for her? If so, what and why?



Follow the Digital Trail

1. What is a digital footprint?

a) A track that animals leave behind

b) Shoes that you buy on the Internet

c) The information about you on the Internet

2. What kind of information is safe to share online?

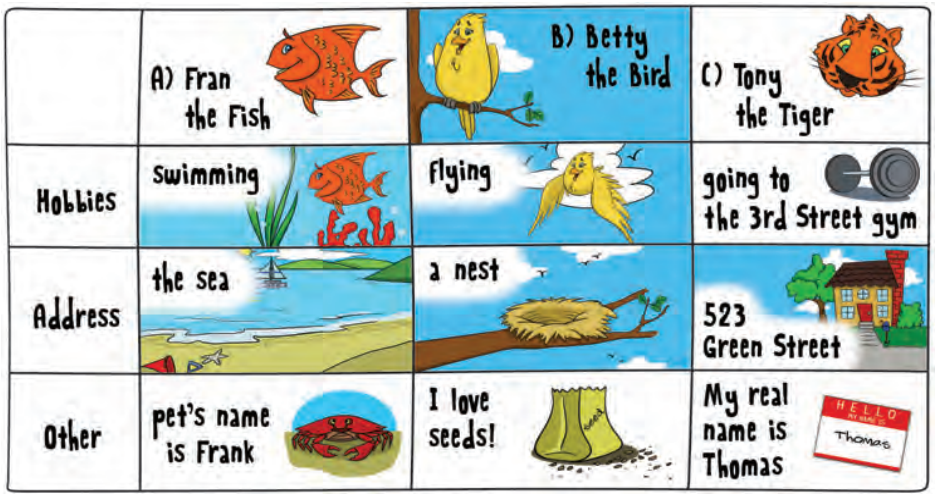
a) Your birth date

b) Your first name or computer username

c) Your address

3. Which animal below has the digital footprint that leaves him or her most unsafe?

HINT: Think about which animal shares the most private information online.



a) Fran the Fish

b) Betty the Bird

c) Tony the Tiger

Follow the Digital Trail

1. What is a digital footprint?

a) A track that animals leave behind

b) Shoes that you buy on the Internet

c) The information about you on the Internet

Answer feedback

The correct answer is **c**. Your digital footprint is the information about you online, such as a news story with your name in it or something that you write online.

2. What kind of information is safe to share online?

a) Your birth date

b) Your first name or computer username

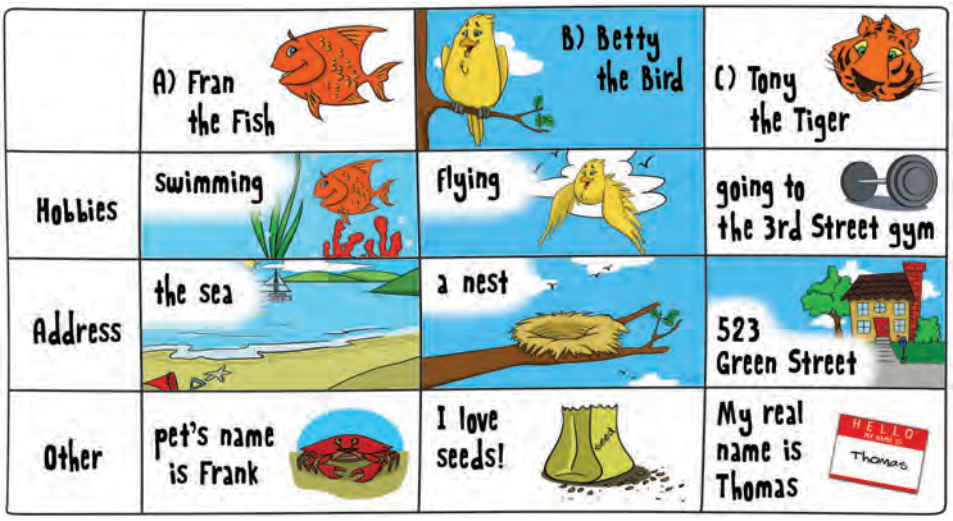
c) Your address

Answer feedback

The correct answer is **b**. It is okay to share your first name or your username online. But sharing your address or birth date could make your information unsafe because other people might use your information to pretend to be you!

3. Which animal below has the digital footprint that leaves him or her most unsafe?

HINT: Think about which animal shares the most private information online.



a) Fran the Fish

b) Betty the Bird

c) Tony the Tiger

Answer feedback

The correct answer is **c**. Tony the Tiger put private information online, like his address, which is not safe. Fran and Betty shared information, but they did not share anything private about themselves.

U

Unplugged

Name: _____ Date: _____

Your Digital Footprint

Staying Safe and Responsible Assessment

CODE

Just because you **can** share something online doesn't mean that you **should**!

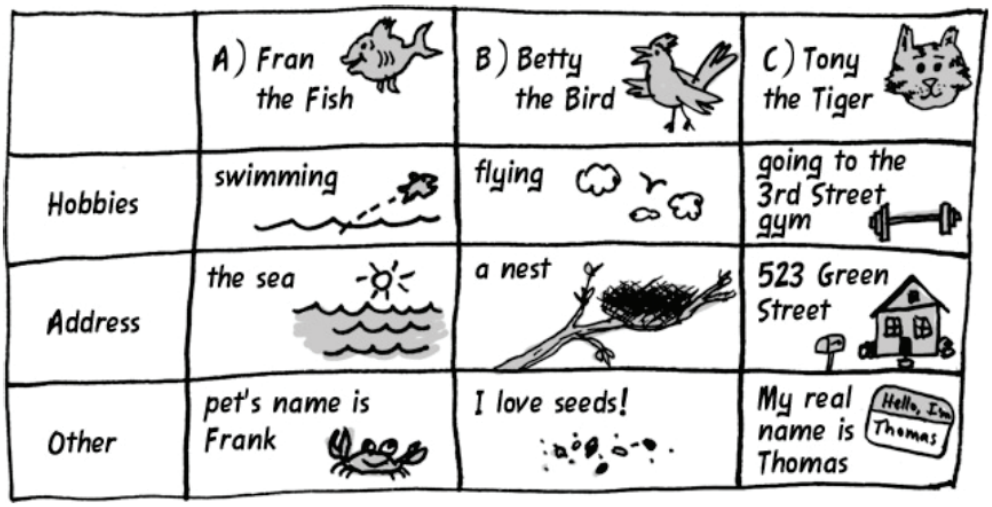
Cross out the information that you should not share online. Use the words that are leftover as the key to what you should find in the word search.

WORDS

1) Your Real Name (NAME)
2) Your Online Name (NICKNAME)
3) Your Address (ADDRESS)
4) Your Email (EMAIL)
5) Your Favorite Color (COLOR)
6) The Last Book you Read (BOOK)
7) Your Credit Card Info (CARD)
8) Your Favorite Band (BAND)
9) Your Phone Number (PHONE)
10) What You Ate Today (FOOD)
11) Your Birthday (BIRTHDAY)

O	H	D	T	X	V	X	N	G	Y
L	X	Q	G	J	U	D	E	M	S
T	B	I	H	T	F	N	B	N	W
H	I	M	D	I	O	I	A	M	G
A	K	S	C	J	O	C	N	O	C
P	E	B	O	M	D	K	A	N	C
P	K	O	L	M	B	N	N	K	O
Y	Y	O	O	G	A	A	A	E	D
V	U	K	R	V	N	M	G	Y	E
R	Z	O	I	F	D	E	C	C	T

Which animal below has the digital footprint that leaves him or her most unsafe?
HINT: Think about which animal shares the most private information online.



Circle One:

A) Fran the Fish

B) Betty the Bird

C) Tony the Tiger

Lesson 6: Programming Unplugged: My Robotic Friends

Algorithms | Debugging | Unplugged

Overview

Using a predefined symbol key, your students will figure out how to guide one another to accomplish specific tasks without using any verbal commands. This segment teaches students the connection between symbols and actions, the difference between an algorithm and a program, and the valuable skill of debugging.

Purpose

This unplugged lesson brings the class together as a team with a simple task to complete: get a "robot" to stack cups in a specific design. Students will work to recognize real world actions as potential instructions in code. The designing of precise instructions will also be practiced, as students work to translate worded instructions into the symbols provided. If problems arise in the code, students should work together to recognize bugs and build solutions.

Agenda

Warm Up (5 min)

- **Introduction**

Main Activity (45 min)

- **My Robotic Friends**

Wrap Up (10 min)

- **Journaling**

Objectives

Students will be able to:

- Gain understanding of the need for precision in coding.
- Learn how to recognize a bug and how to debug the malfunctioning code.

Preparation

- [ ] Read **My Robotic Friends - Teacher Prep Guide**.
- [ ] Print out one **My Robotic Friends - Symbol Key** per group. This is "code" to be used.
- [ ] **Paper Trapezoid Template - Manipulatives** are provided if your class is not going to use cups.
- [ ] Print out one set of **Stacking Cups Ideas - Teacher Prep Guide** per group.
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **My Robotic Friends** - Video
- **My Robotic Friends** - Symbol Key (**PDF** | **DOCX**)
- **Paper Trapezoid Template** - Manipulatives (**PDF** | **DOCX**)
- **Stacking Cups Ideas** - Teacher Prep Guide (**PDF** | **DOCX**)
- **My Robotic Friends** - Teacher Prep Guide (**PDF** | **DOCX**)

Vocabulary

- **Algorithm** - A precise sequence of instructions for processes that can be executed by a computer
- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in your algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

Warm Up (5 min)

Introduction

Start by asking the class if anyone has heard of robotics. Has anyone seen a robot or touched one? Does a robot really “hear” you speak? Does it really “understand” what you say? The answer to the last question is: “Not the same way that a person does.”

Robots operate off of instructions, specific sets of things that they have been preprogrammed to do. In order to accomplish a task, a robot needs to have a series of instructions (sometimes called an algorithm) that it can read. Today, we are going to learn what it takes to make that happen.

If you feel that there is time, it might be helpful to do a quick example. Page 6 and 7 of **My Robotic Friends - Teacher Prep Guide** describe how to do a simple example before the main activity. This example could be up to 10 minutes in length.

Main Activity (45 min)

My Robotic Friends

Display a copy of the **My Robotic Friends - Symbol Key** (or write the symbols on the board). Step to the side and tell the class that these will be the only six symbols that they will be using for this exercise. For this task, they will instruct their “robot” friend to build a specific cup stack using only the arrows listed on the **My Robotic Friends - Symbol Key**.

Lesson Tip:

While the robot is working on the stack make sure that the class knows:

- Programmers are not allowed to talk when the robot is working. This includes blurting out answers or pointing out when the robot has done something wrong.
- Programmers should raise their hand if they see a bug.

1. Pick a "robot" for the class (try to choose a student that feels confident after the warm-up.) Ask this "robot" to leave the classroom until they are called back in.
2. Display **Stacking Cups Ideas - Teacher Prep Guide** to the rest of the class. Have the class vote and choose which idea they would like the robot to do. Try to push for an easier idea for the first time, then choose a more complex design later on.
3. Let the class discuss how the stack should be built, then ask the class to translate the algorithm into the symbols. Write down the symbol algorithm somewhere for the robot to use later.
4. When the class has decided on the algorithm, ask the robot to come back in. We recommend continuing to display the **My Robotic Friends - Symbol Key** while the robot is building the stack so the student robot remembers what each command means.

If a student sees a bug and raises their hand, have the robot finish the instructions to the best of their ability. Afterward, have the students discuss the potential bug and come up with a solution. Continue repeating until the stack is built properly.

Lesson Tip:

This activity can be done in small groups, once your class feels comfortable.

Once the stack is built, you can choose to repeat the activity again with another student robot.

Wrap Up (10 min)

Journaling

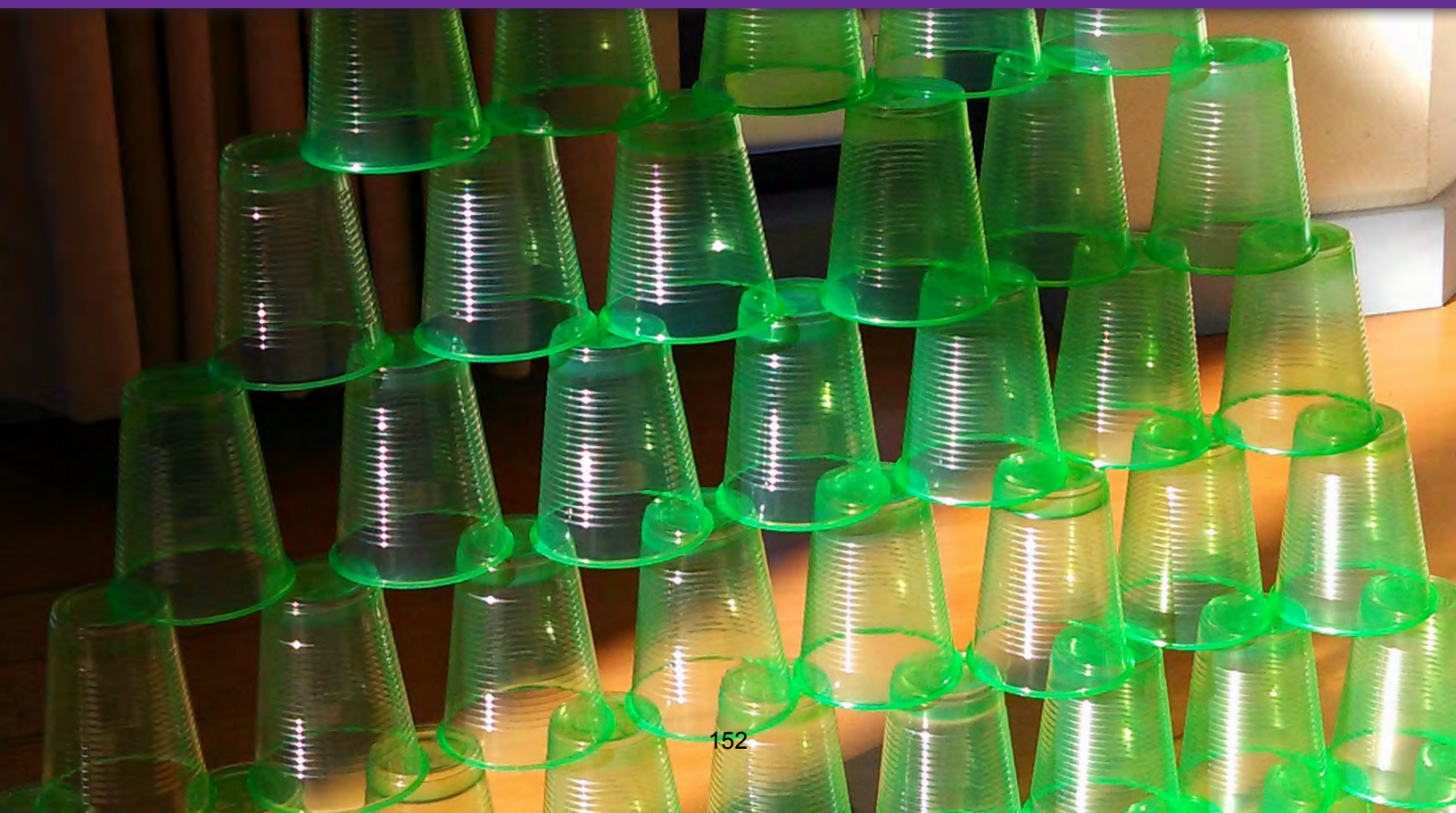
Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw a stack of cups that the robot made today.
- Draw a stack of cups that you would like a robot to make someday!

MY ROBOTIC FRIENDS

Modified for Code.org CS Fundamentals



Disclaimer
Neither Thinkersmith nor any other party involved in the creation of this curriculum can be held responsible for damage, mishap, or injury incurred because of this lesson. Adult supervision and supervised caution is recommended at all times. When necessary, every effort has been made to locate copyright and permission information.

Lesson 3: My Robotic Friends

Main Goal: Highlight programming techniques and illustrate the need for functions.

Overview: Using a predefined "Robot Vocabulary" your students will figure out how to guide one another to accomplish specific tasks without discussing them first. This segment teaches students the connection between symbols and actions, as well as the valuable skill of ***debugging***.

If time allows, there is an option to introduce ***functions*** at the end of the lesson.

Objectives: **Students will**

- Learn to convert real-world activities into instructions
- Gain practice ***coding*** instructions with symbols
- Gain understanding of the need for precision in coding
- Gain practice debugging malfunctioning code
- Understand the usefulness of functions and ***parameters*** (grades 7+)

Materials and Preparation:

Estimated lesson time: 1 hour
Estimated prep time: 10 min

Materials

- Symbol Key (1 per group)
- Cup Stack Pack (1 per group)
- Disposable Cups or Paper Trapezoids (6 or more per group)
- Blank paper or note cards (1 per person)
- Writing Instrument (1 per person)

Preparation

- Print out one Symbol Key for each group
- Print a Cup Stack Pack for each group
- Cut trapezoids from Paper Trapezoid template if not using cups
- Stack cups or trapezoids in designated area away from groups (Robot Library)

Key Lesson Vocabulary:

Algorithm - A series of instructions on how to accomplish a task

Coding - Transforming actions into a symbolic language

Debugging - Finding and fixing issues in code

Function - A piece of code that can be called over and over

Parameters - Extra bits of information that you can pass into a function to customize it

Lesson Plan

Introduce: Start by asking the class if anyone has heard of robotics. Has anyone seen a robot or touched one? Does a robot really "hear" you speak? Does it really "understand" what you say? The answer to the last question is:
"Not the same way that a person does."

Robots operate off of "instructions", specific sets of things that they have been preprogrammed to do. In order to accomplish a task, a robot needs to have a series of instructions (sometimes called an algorithm) that it can run. Today, we are going to learn what it takes to make that happen.

Kickstart: Pull out a copy of the Symbol Key (or write the symbols on the board). Step to the side and tell the class that these will be the only six symbols that they will be using for this exercise. For this task, they will instruct their "robot" to build a specific cup stack using only these arrows:

- ↑ - Pick Up Cup
- ↓ - Put Down Cup
- → - Move 1/2 Cup Width Forward
- ← - Move 1/2 Cup Width Backward
- ↻ - Turn Cup Right 90°
- ↺ - Turn Cup Left 90°

Adjustments:

Grades K-3

- Try this lesson all together as one class. Let the students shout directions for the teacher to write down.
- Have a class assistant leave the room during programming, then return to perform the finished code.
- If there is time, switch. Have the assistant write the instructions from the class and have the teacher perform them.

Grades 4-6

- Adjust group sizes between three and five, depending on personality of class.
- Expect each student to want a turn, this will likely use the entire hour.

Grades 7+

- Limit groups to four students, three is ideal.
- Students generally complete the full round of turns with plenty of time to include the supplement section on functions.

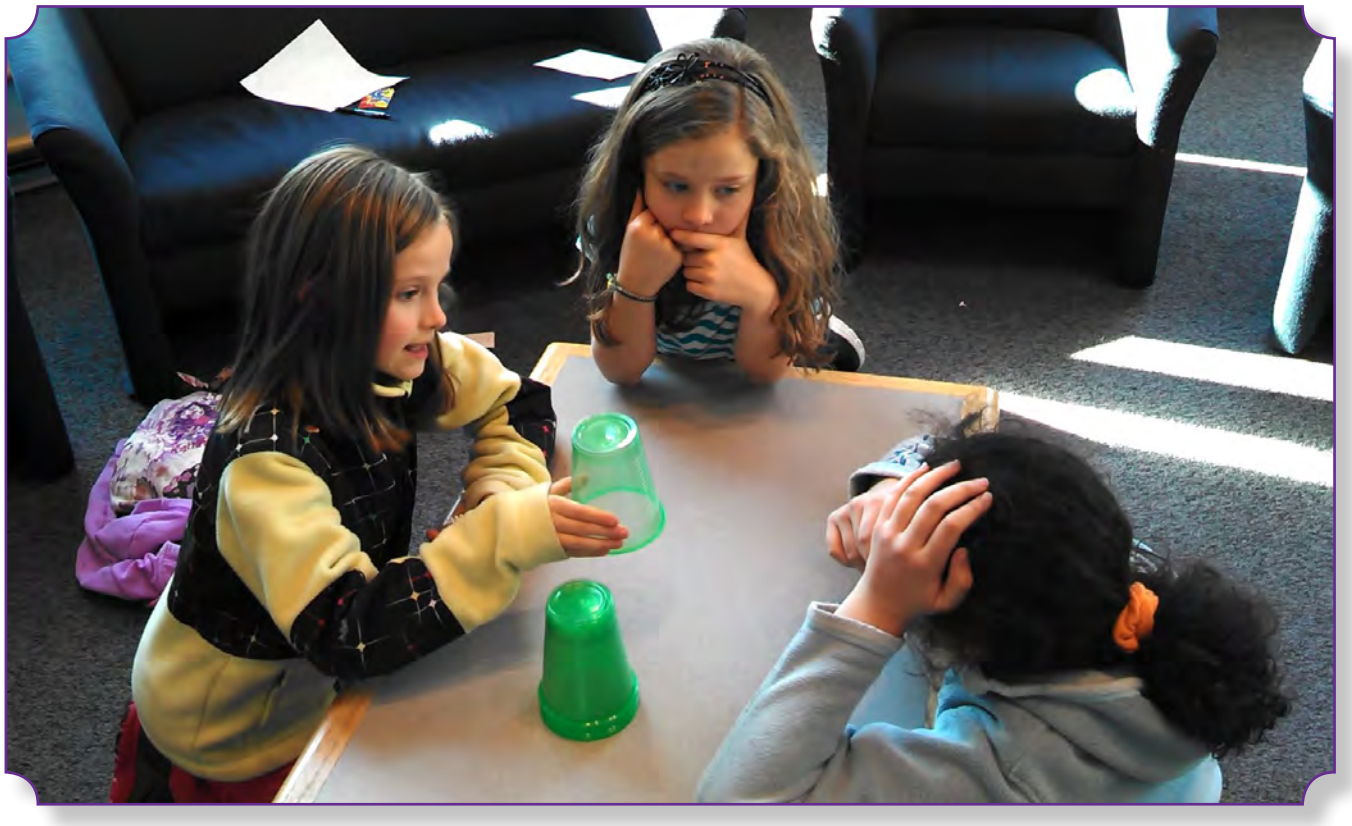
Steps:

1. Choose one "Robot" per team.
2. Send robot to "Robot Library" while the "programmers" code.
3. Choose one image from the Cup Stack Pack for each group.
4. Groups will create an algorithm for how the robot should build the selected stack.
5. Coders will translate their algorithm to arrows, as described in Symbol Key.
6. When programmers have finished coding their stack they can retrieve their robot.
7. Upon return, the robot reads the symbols from the cards and translates them back in to movements.
8. The group should watch for incorrect movements, then work together to debug their program before asking the robot to re-run it.

Rules:

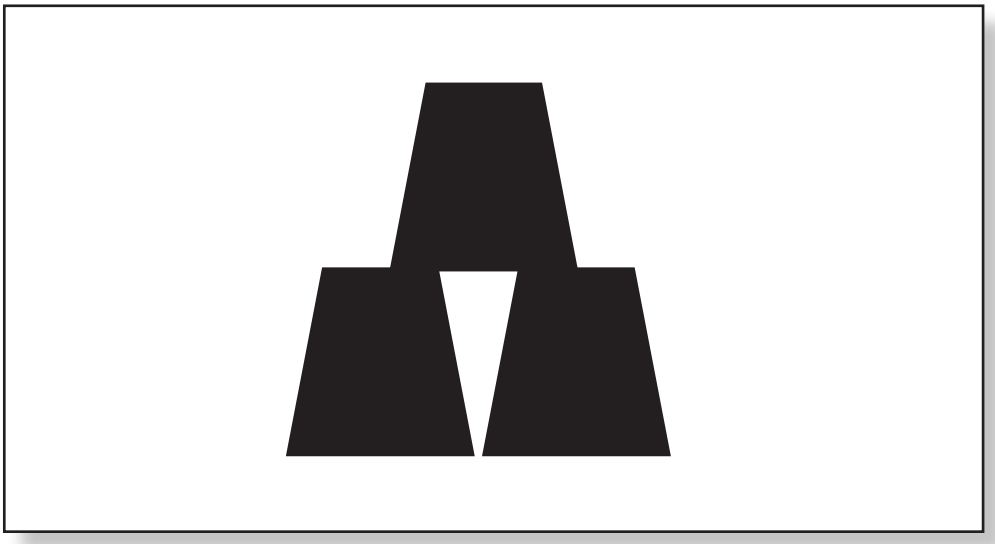
1. Coders should translate all moves using *only* the six arrows suggested.
2. Cups should remain with the robot, not provided to programmers during coding.
3. Once robots are back with their groups, there should be no talking out loud.

If your student asks about rules that haven't been defined above, you can either define them according to your exercise, or ask them to define that rule within their own group.



Example

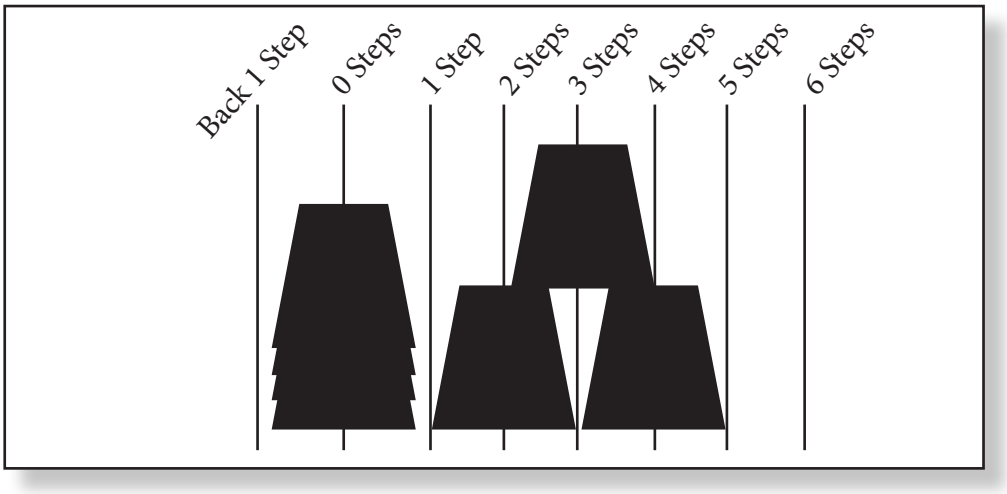
Beginning: It can be helpful to go over an example as a class. There is one cup stack in the pack that includes only three cups, that is the sample card. Hold it up for the class and walk them through the exercise.



3 Cup Stack from Cup Stack Pack

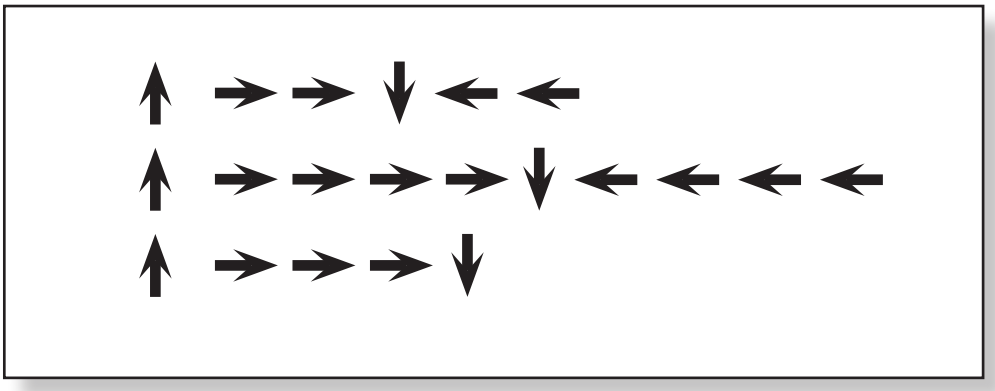
Place your stack of cups on the table where everyone can see them. Ask the class to instruct you on the first thing to do. The correct answer is "pick up cup". When you pick up each cup, note that the cup should automatically rise above the highest cup already in the stack.

With your hand still in the air, ask for the next move. You may have to remind the class a time or two that one step forward is only half the width of a cup.



Step Guide

Middle: Once you've placed a single cup, transition back to the blackboard (or document camera) and challenge the class to help you write the symbols on the board so that you can "run the program" later. One possible solution looks like this:



One Solution for 3 Cup Stack

Completion: With the program written down for the class to see, you can call a volunteer to "run" it, or you can run the program yourself. Say the arrows out loud as you move the cups into place. For example, the program above would be pronounced:

"Pick up cup", "Step forward", "Step forward", "Put down cup"
"Step backward", "Step backward"

"Pick up cup", "Step forward", "Step forward", "Step forward",
"Step forward", "Put down cup", "Step backward", "Step backward",
"Step backward", "Step backward"

"Pick up cup", "Step forward", "Step forward", "Step forward",
"Put down cup"

The Exercise

Group Up: Group the students appropriately for their age as described on page 2. The goal is to have enough programmers in each group that the group is never entirely lost.

Robot: Choose one "robot" in each group to go hang out in the "robot library". This should be a location far enough away from the groups that no robot can find out what Cup Stack Card their programmers are working with. Robots can use their time in the library to practice cup stacking and to ask for clarification on rules.

Program: Each group of programmers should be handed one Cup Stack Card at a time. They can then begin to figure out the algorithm for their stack. How many cups will they need? How many steps for the first cup? The second? Are any cups upside down? How do you get the robot to flip a cup?

Once these questions are answered, the programmers can use the symbols to write their code on the blank paper or a note card. The programmers should review their code to see if it makes sense for the stack before checking their robot out of the robot library. Do not let students use symbols that are not on the Key (such as numbers.)

Run Code: Now that the robot is back with the group, everyone should be silent. The groups should not attempt to use words or gestures to influence their robot's behavior. The robot should only operate according to what the arrows tell them to do.

If the group finds a mistake, they are allowed to halt the program, check the robot back into the library, and fix the error before bringing the robot back to complete the challenge.

Repeat: Each time a group solves a challenge, they should choose a new robot to head to the library, and the group should be given a new (preferably more difficult Cup Stack Card.)

This can continue either until time is done, all group members have been robots, or the cards have become difficult enough to warrant a discussion about functions.

Tip: If the lesson is still going strong, but the groups begin to run out of Cup Stack Cards, challenge them to create their own stack drawings.

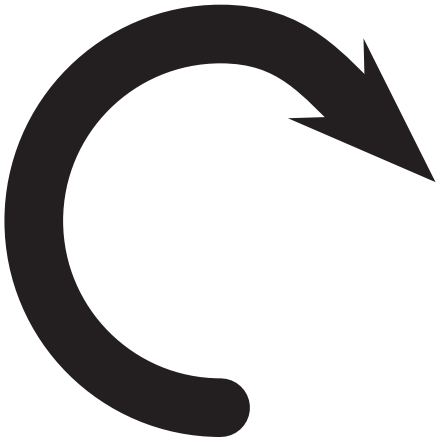
Symbol Key

Pick Up Cup

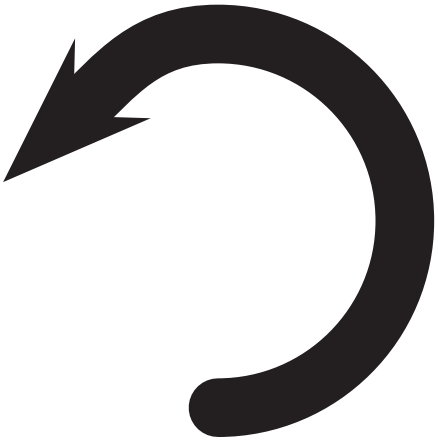
Put Down Cup

Step Forward

Step Backward



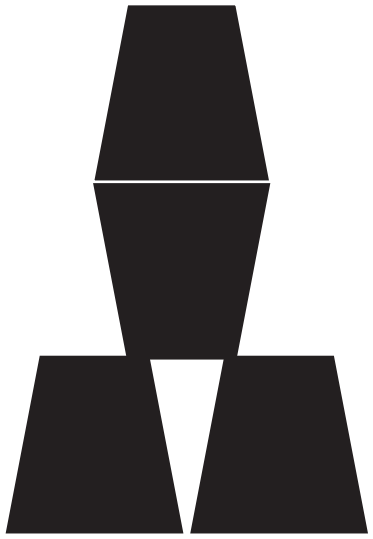
Turn Cup Right 90°

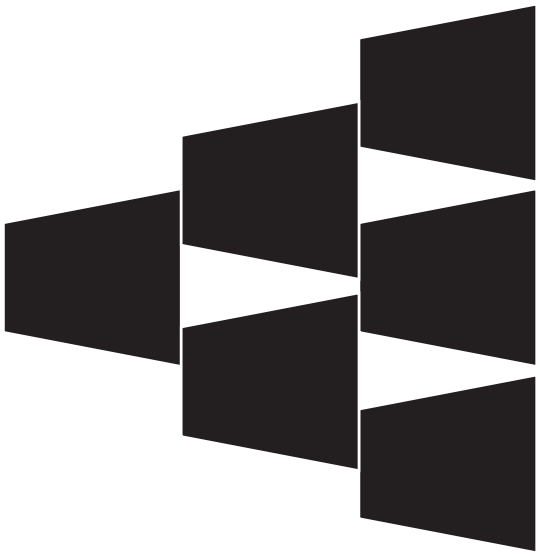


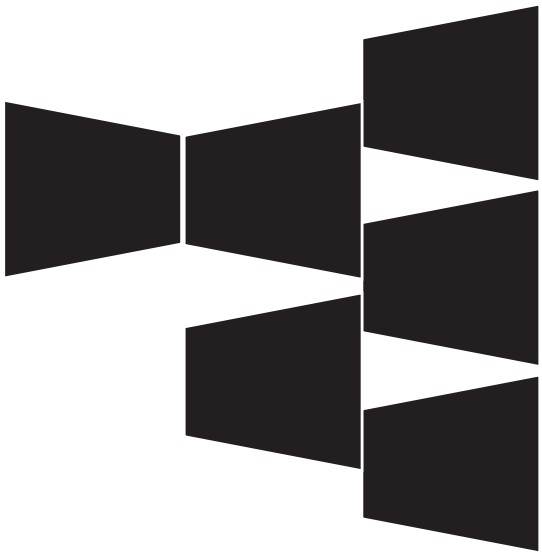
Turn Cup Left 90°

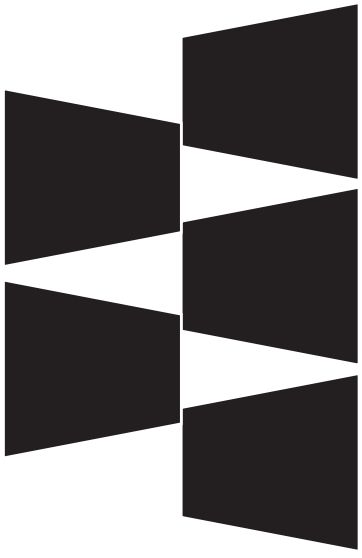
Cup Stack Pack™

THINKERSMITH™

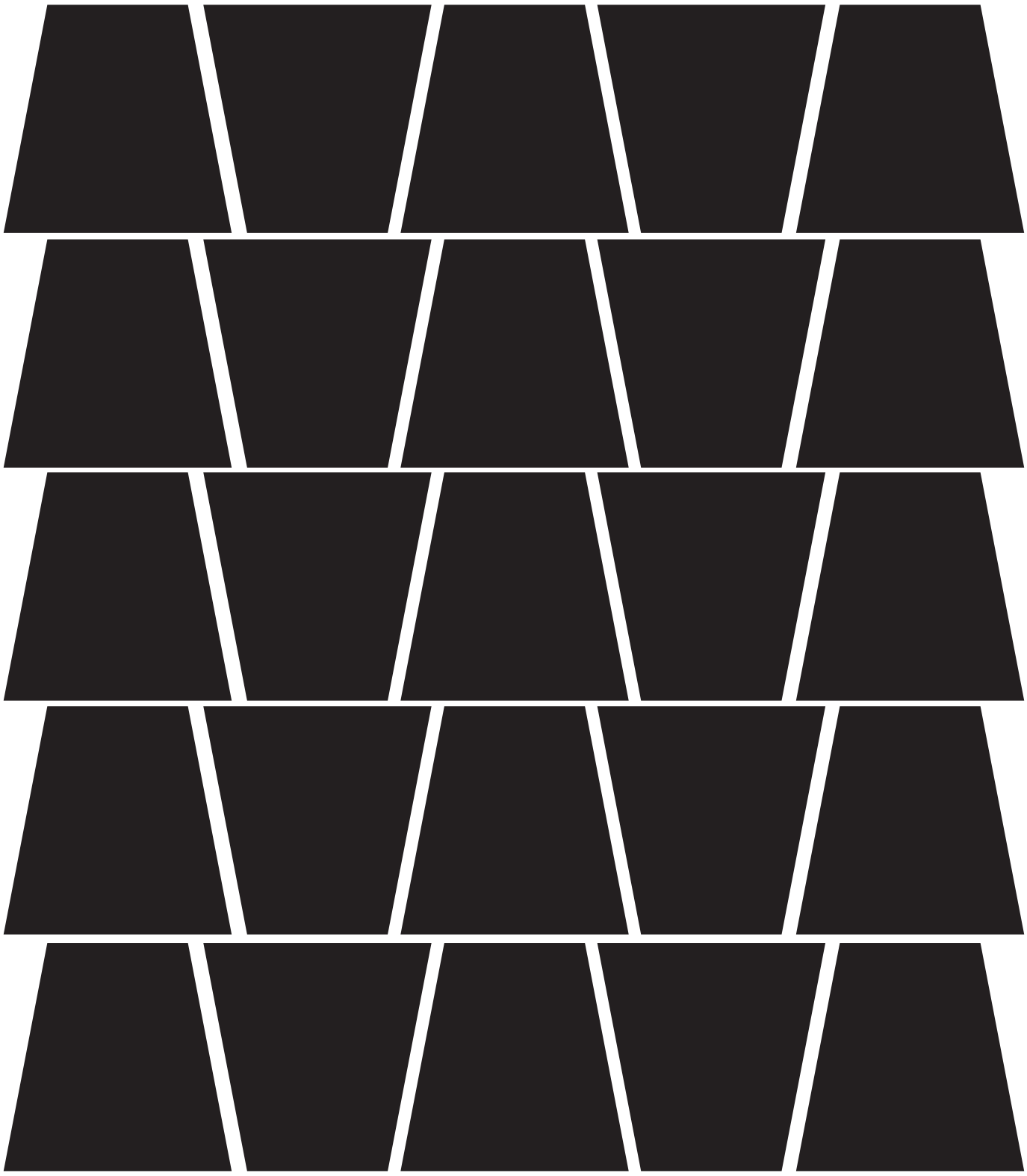








Paper Trapezoids



To cut quickly:

First cut in horizontal strips, then snip along lines to make trapezoids.

Lesson 9: Loops Unplugged: My Loopy Robotic Friends

Unplugged | Loop | Repeat

Overview

Here, students learn the simplicity and utility of loops by “programming” their friends using the language from ‘My Robotic Friends’. Once loops are introduced, students will find that they can build bigger structures faster.

Purpose

This lesson serves as an introduction to loops. *Loops* allow for students to simplify their code by grouping commands that need to be repeated. Students will develop critical thinking skills by noticing repetition in the movements of classmates and determining how many times their code needs to be looped. By seeing 'My Robotic Friends' again, students will be able to relate old concepts (such as sequencing) to the new concept of loops.

Agenda

Warm Up (10 - 15 min)

- **My Robotic Friends Review**
- **Loops to the Rescue**

Main Activity (15 - 20 min)

- **Looping with My Robotic Friends**

Wrap Up (8 min)

- **Journaling**

Extension Activities

Objectives

Students will be able to:

- Identify repetitive code and convert a series of multiple actions into a single loop.
- Decode loops into a series of multiple actions.

Preparation

- [ ] Print one **My Robotic Friends Loops - Teacher Prep Guide** for each group.
- [ ] Acquire up to 20 paper cups for each group.
- [ ] Review **My Robotic Friends - Video**.
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **My Robotic Friends** - Video
- **My Robotic Friends Loops** - Teacher Prep Guide
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again

Teaching Guide

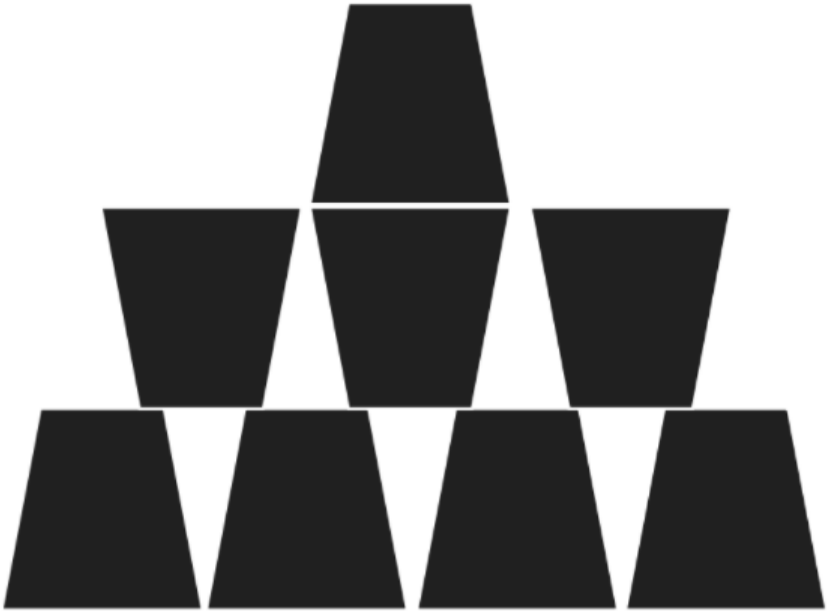
Warm Up (10 - 15 min)

My Robotic Friends Review

Goal: This review will refresh the students' minds about how quickly programs for the My Robotic Friends activity can get intense.

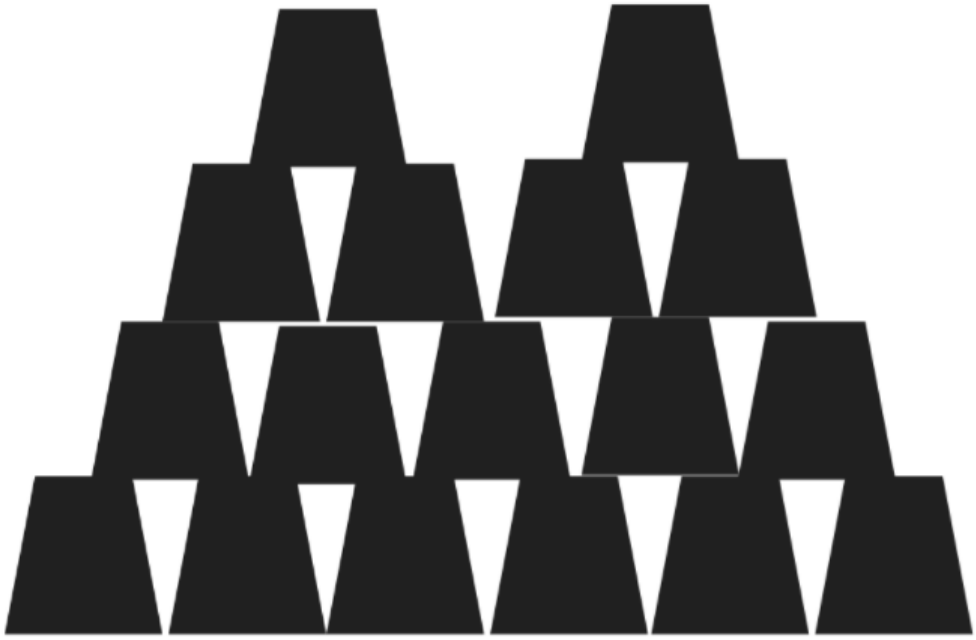
With the class together as a group, pull an easy puzzle from the My Robotic Friends Cup Stack Pack and program with each other as a reminder of rules and terminology.

Next, pull a puzzle that's slightly harder, but also requires a lot of steps like the one below.



Ask a volunteer (or a group of volunteers) to come forward to help program this one on the board. If you make them stick strictly to the "no symbols other than those on the key" rule, it will probably take a while!

Now, bring up this image:



What is the reaction of the class?

Loops to the Rescue

Give students the opportunity to brainstorm shorter ways to relay the code that they're about to create. (This bit can be skipped over if your students start saying things like: "Move forward 6 times." Since that will open the discussion about how to show "six times" with symbols.)

Once students have put together the idea of "repeating" code, give them the vocabulary around it. Make sure to share with them that often the terms "repeat something" and "loop something" will be used interchangeably on Code.Org.

Main Activity (15 - 20 min)

Looping with My Robotic Friends

Goal: This activity will allow students to gain practice spotting areas that repeat loops can be used, as well as places where they need to expand programs that utilize loops.

Practice Makes Perfect

Take the program from one of your previous cup stacks and display it for the class. Have them help you find places that the same arrows repeat, uninterrupted, multiple times. Have students count the number of times those arrows repeat and give you the final tally.

Circle the first arrow in that line, write the number of loops near that circle, then cross out the rest of the arrows.

Repeat this until the entire program has been shortened, then re-write the program in a way where students can see how much more simple the resulting instructions are.

Looping with My Robotic Friends

Now that students have a new tool in their toolbox, they should be able to start finding success on new (and more difficult) cup stacks.

Set students to task with the cards from the more difficult My Robotic Friends Loops Packet and see how they do. You can either continue to work together, or let students work in small groups -- whichever is best for your classroom.

Teaching Tips:

Be sure to keep your eyes open for students using loops. Try to avoid correcting their overall algorithms, but feel free to point out instructions that could be shortened by using a repeat circle.

Watch students as they run through the code. Are there any bugs? Use the debugging questions to help them find a solution.

- What does it do?
- What is it supposed to do?
- What does that tell you?
- Does it work at the first step?
- Does it work at the second step?
- Where does it stop working?

Wrap Up (8 min)

Journaling

Goal: Allow students to reflect on the activity that they just experienced.

Flash Chat:

Here are some possible topics:

- Do you feel like loops make programming easier or harder?
- What other kinds of things in life do we repeat?
  - Eating - put food in mouth, chew 20 times
  - Brushing hair - brush through hair 35 times
  - Routines - Wake up, go to school, come home, go to bed

Journal Prompts:

- Journal time! Ask students to draw a feeling face in the corner of their journal page to remind them how they felt about this lesson.
- Have the students write or draw something in their journal that will remind them later what loops are. This can come from a prompt like:
  - What does "repeat" mean to you?
  - Draw a picture of you repeating something.

Extension Activities

- Have students draw their own cup stacking creations for someone else to code.
- Provide students with algorithms that utilize repeats, then have them expand the program back out to a full step-by-step version.

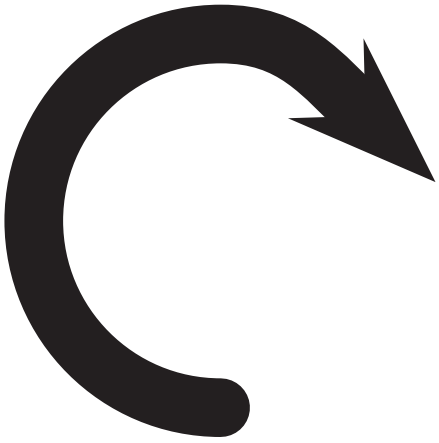
Symbol Key

Pick Up Cup

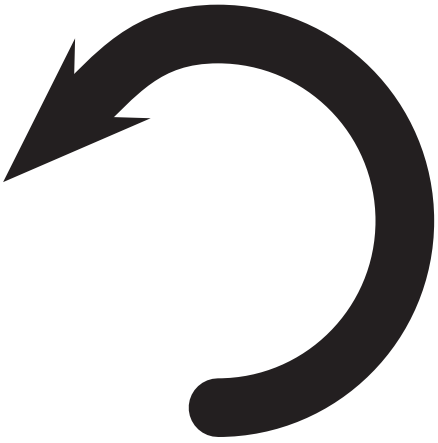
Put Down Cup

Step Forward

Step Backward



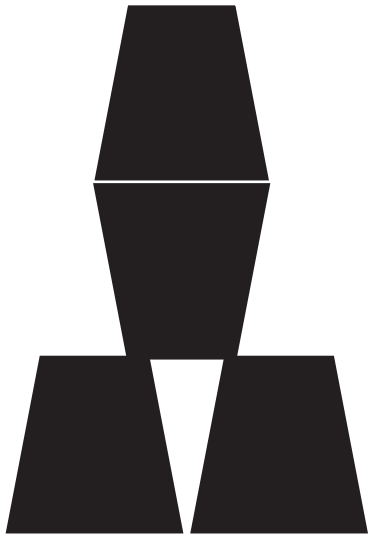
Turn Cup Right 90°

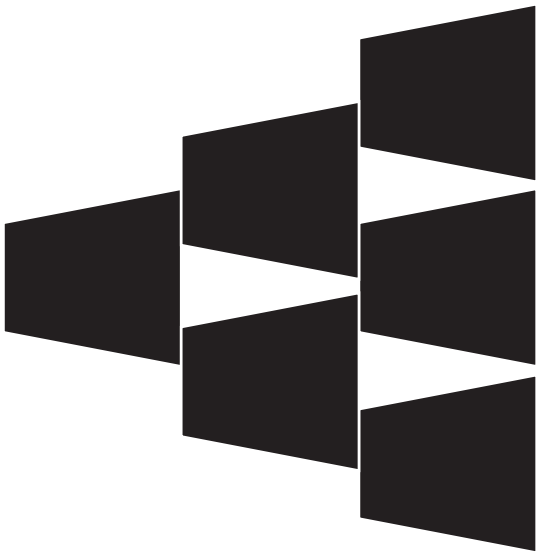


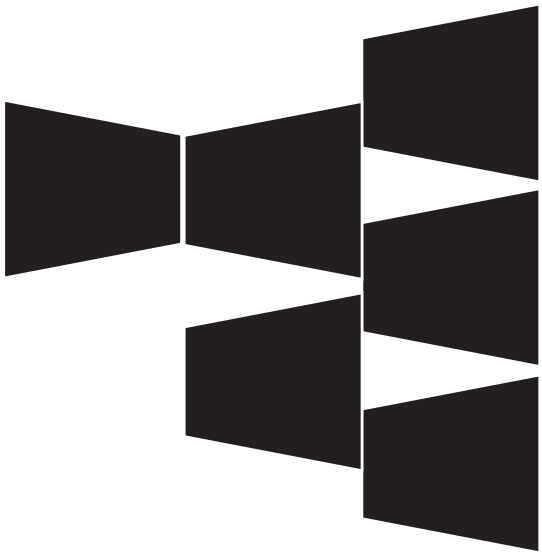
Turn Cup Left 90°

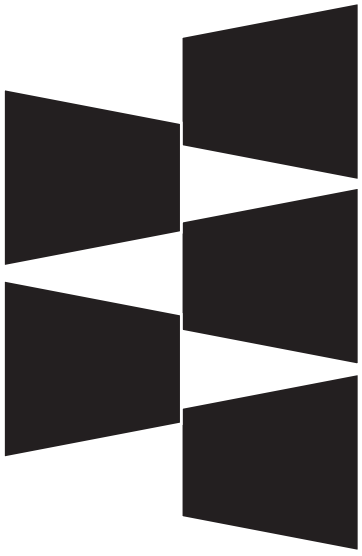
Cup Stack Pack™

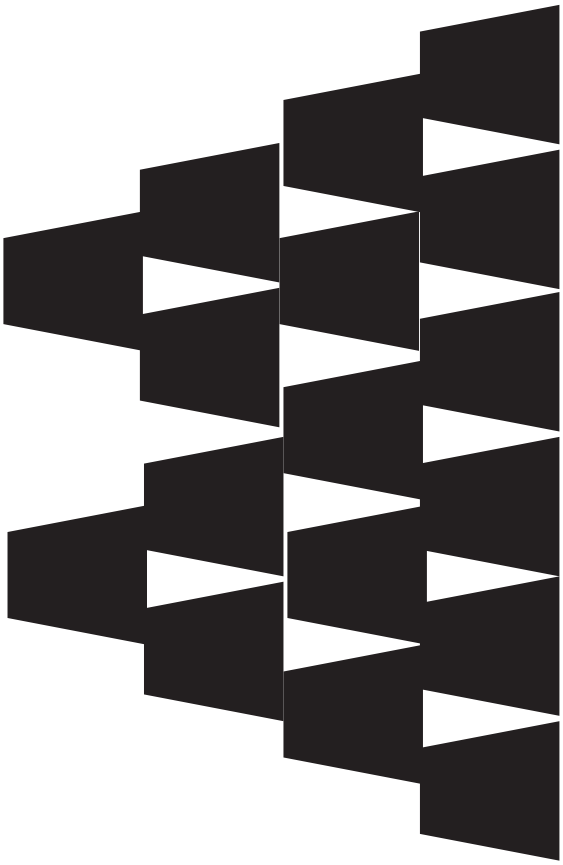
THINKERSMITH™

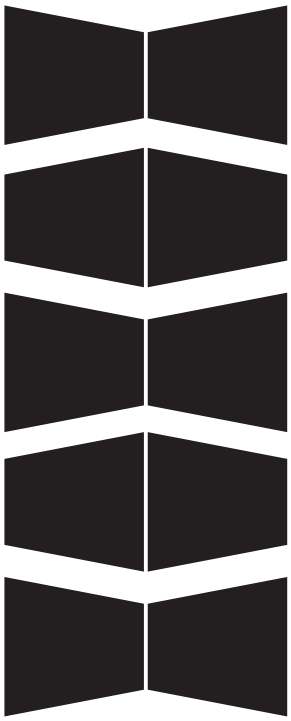




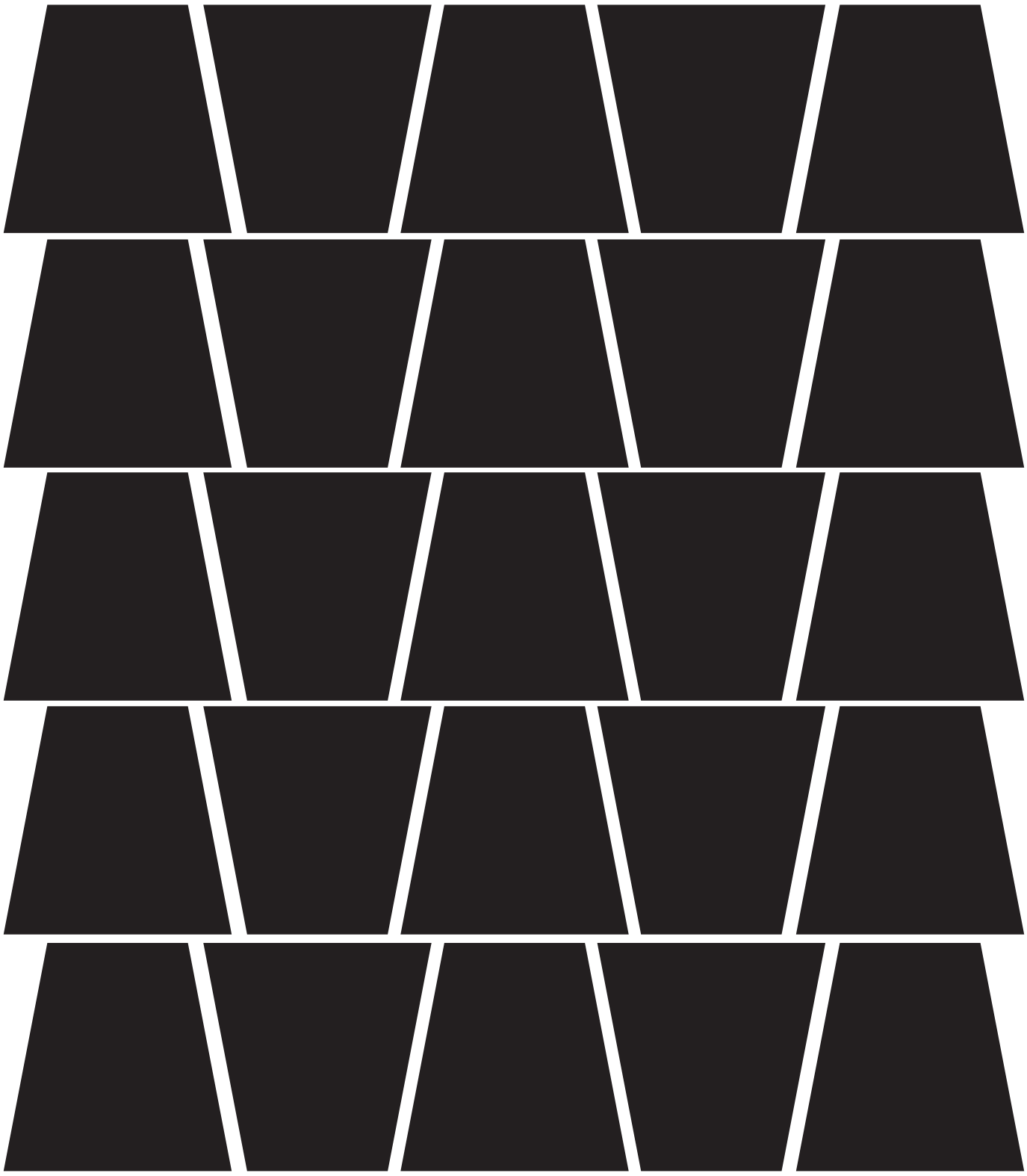








Paper Trapezoids



To cut quickly:

First cut in horizontal strips, then snip along lines to make trapezoids.

Lesson 12: Events Unplugged: The Big Event

Event | Unplugged

Overview

Events are a great way to add variety to a pre-written algorithm. Sometimes you want your program to be able to respond to the user exactly when the user wants it to. That is what events are for.

Purpose

Today, students will learn to distinguish events from actions. The students will see activities interrupted by having a 'button' pressed on a paper remote. When seeing this *event*, the class will react with a unique action. Events are widely used in programming and should be easily recognizable after this lesson.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **A Series of Events**

Main Activity (15 min)

- **The Big Event**

Wrap Up (10 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (10 min)

- **CSF The Big Event Activity - Assessment**

Extended Learning

Objectives

Students will be able to:

- Repeat commands given by an instructor.
- Recognize actions of the teacher as signals to initiate commands.
- Practice differentiating pre-defined actions and event-driven ones.

Preparation

- [ ] Watch the **The Big Event - Teacher Video**.
- [ ] Print one **CSF The Big Event Activity - Worksheet**.
- [ ] Print one **CSF The Big Event Activity - Assessment** for each student.
- [ ] Make sure each student has a **Think Spot Journal**.

Links

For the Teacher

- **The Big Event** - Teacher Video
- **CSF The Big Event Activity** - Worksheet
- **CSF The Big Event Activity** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Event** - An action that causes something to happen.

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important vocabulary word:

Event - Say it with me: E-vent

An action that causes something to happen

A Series of Events

- Prep your class to answer a question:
  - "I'm going to ask you a question. I want you to raise your hand if you want me to call on you for the answer."
  - Ask a simple question that most of your students should be able to answer, such as:
    - How many thumbs do I have?
    - What is bigger, a bird or a horse?
  - Call on a student who has their hand raised and let them give their answer.
  - Upon finishing that display, ask the class how you knew that the student wanted you to call on them.
    - Your class will likely mention the raising of the hand.
  - Explain to everyone that when students raise their hand, it is an "event" that causes you to know that they want to be called on.
- Ask the class if they can think of any other events that give signals.
  - You may need to remind them that you're not talking about an event like a birthday party or a field trip.
  - If they have trouble, you can remind them that an event is an action that causes something to happen.
    - What about an alarm clock going off? What does that make happen?
    - What about pressing "Start" on the microwave? What does that do?
    - What about pressing the power button on your tv remote?
- Today, we're going to create programs with events.

Main Activity (15 min)

The Big Event

- Do you remember helping the Red, the Angry Bird find the pig?
  - In that exercise, you knew in advance exactly where you wanted Red to end up, so you could make a program that took the bird from start to finish without any interruptions.
  - In most real programs, we can't do that because we want to have options, depending on what the user needs.
    - Say that I only want my character to move when my finger is on the screen of my phone. I would need to program the character to only move when I put my finger on the screen of my phone.
    - Putting my finger on the screen would then become an "event" that tells my character to move.

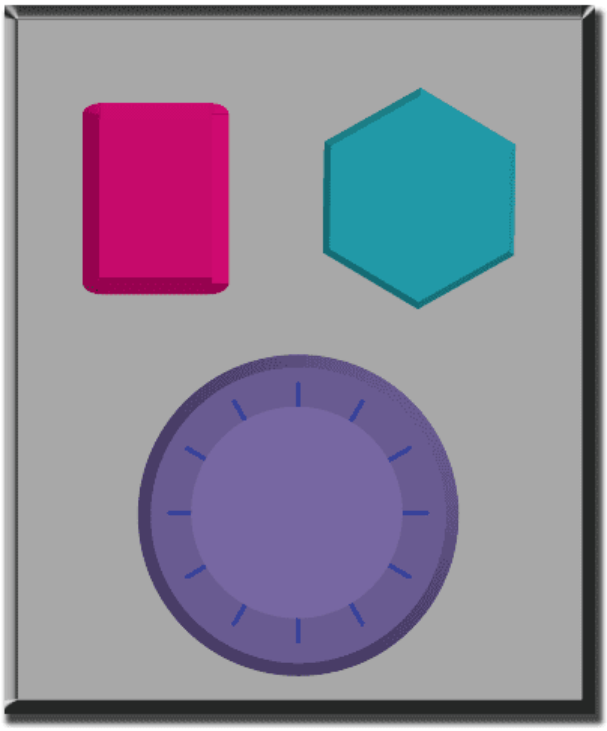
Lesson Tip

If your students seem confused, talk about their favorite games and all of the ways that they let the characters know what they're supposed to do. Point out how the game would be really boring if it ran from start to finish without any events required.

In earlier lessons, we created algorithms that allowed us to control a friend or bird for several steps at a time. It was fun and useful, but what happens when you don't know everything that you want your friend to do in advance? This is where events come in!

Directions:

- Project the Event Controller onto your classroom screen.



- Decide with your class what each button does. We suggest:
  - Pink Button -> Say “Wooooo!”
  - Teal Button -> “Yeah!”
  - Purple Dial -> “Boom!”
- Practice tapping the buttons on the overhead and having your class react.
- Add some button sequences into the mix and have the students try to keep up with their sounds.
- Let your class know that every time you push a button, it is an “event” that lets them know what they are expected to do next.
- Get the class started on a planned task before interrupting them again with the buttons. We suggest:
  - Counting to 10
  - Singing “Old MacDonald”
- Once their plan is underway, interject button presses sporadically.
- Continue the blend until they understand the difference between actions that are guided by a plan and those that are event driven.

Wrap Up (10 min)

Flash Chat: What did we learn?

- Why do we need to be able to handle events in a program?
- What are some other kinds of events that you can think of?

Journaling

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- Draw an event that caused an action today.
- Draw an action that was caused by an event that happened today.

Assessment (10 min)

CSF The Big Event Activity - Assessment

- Hand out the assessment activity and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

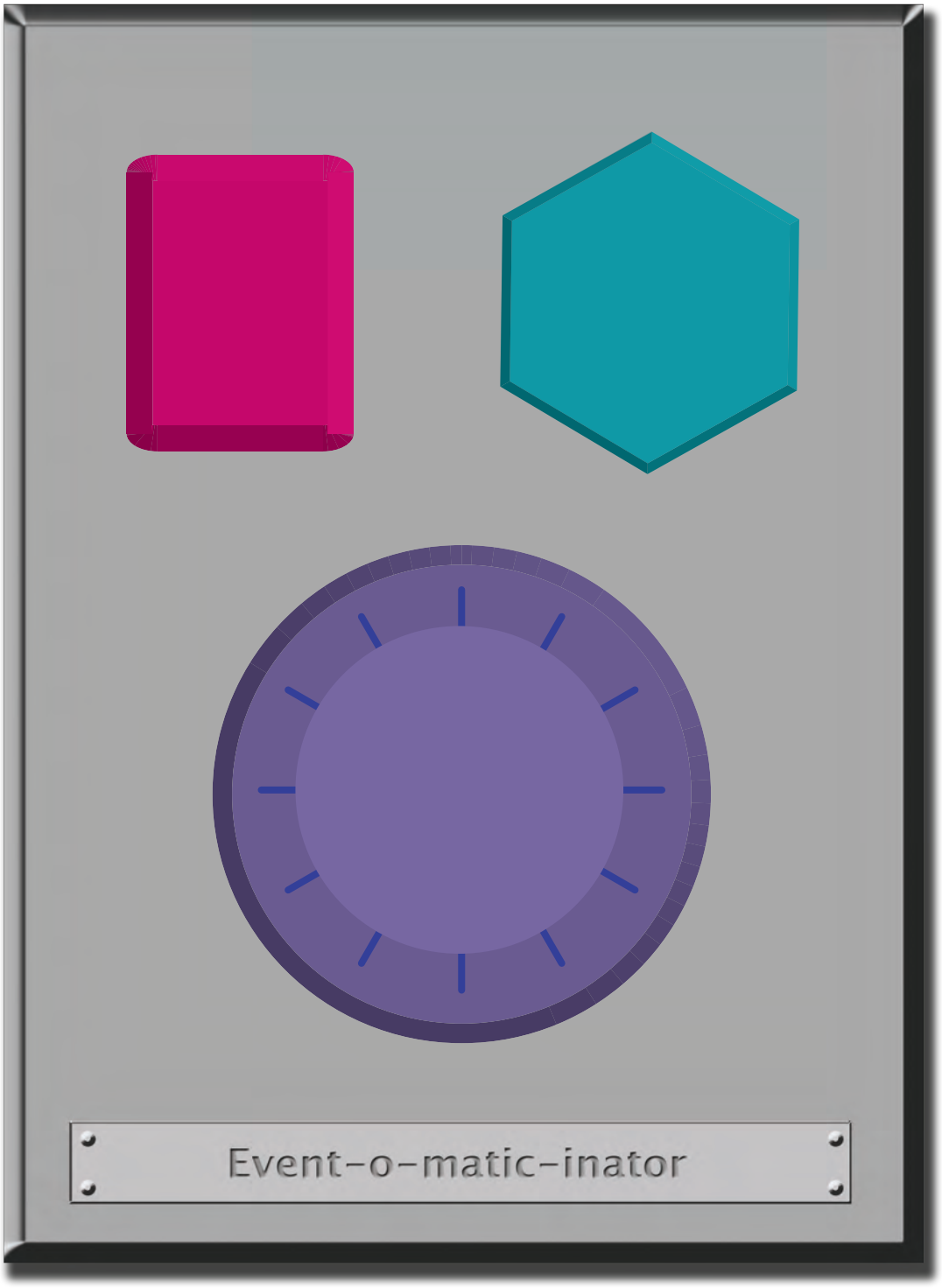
Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

One Person's Event is Another One's Reaction

Assign each student an event to watch out for, and an appropriate reaction to that event. Chain the actions so that each child's reaction becomes an event that triggers the reaction of another student. Keep assigning until everyone has something to do and everyone makes someone react.

Eventopalooza

Break the class up into groups. Using the Events Controller, assign each group a different reaction to the same button. Do this for all three buttons, then watch the chaos!


Event-o-matic-inator

U

Unplugged

Name: _____ Date: _____

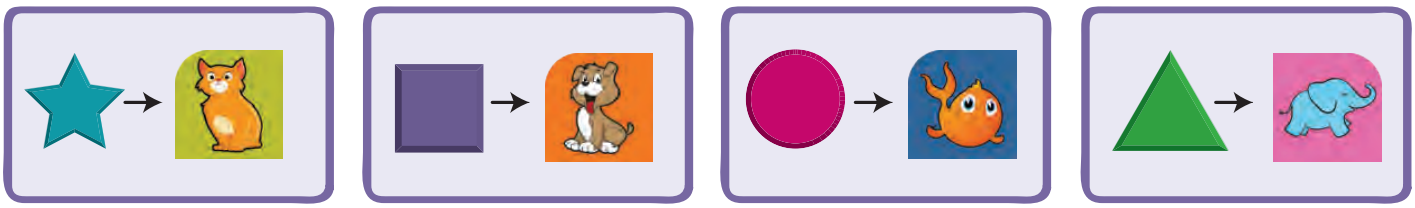
The Big Event

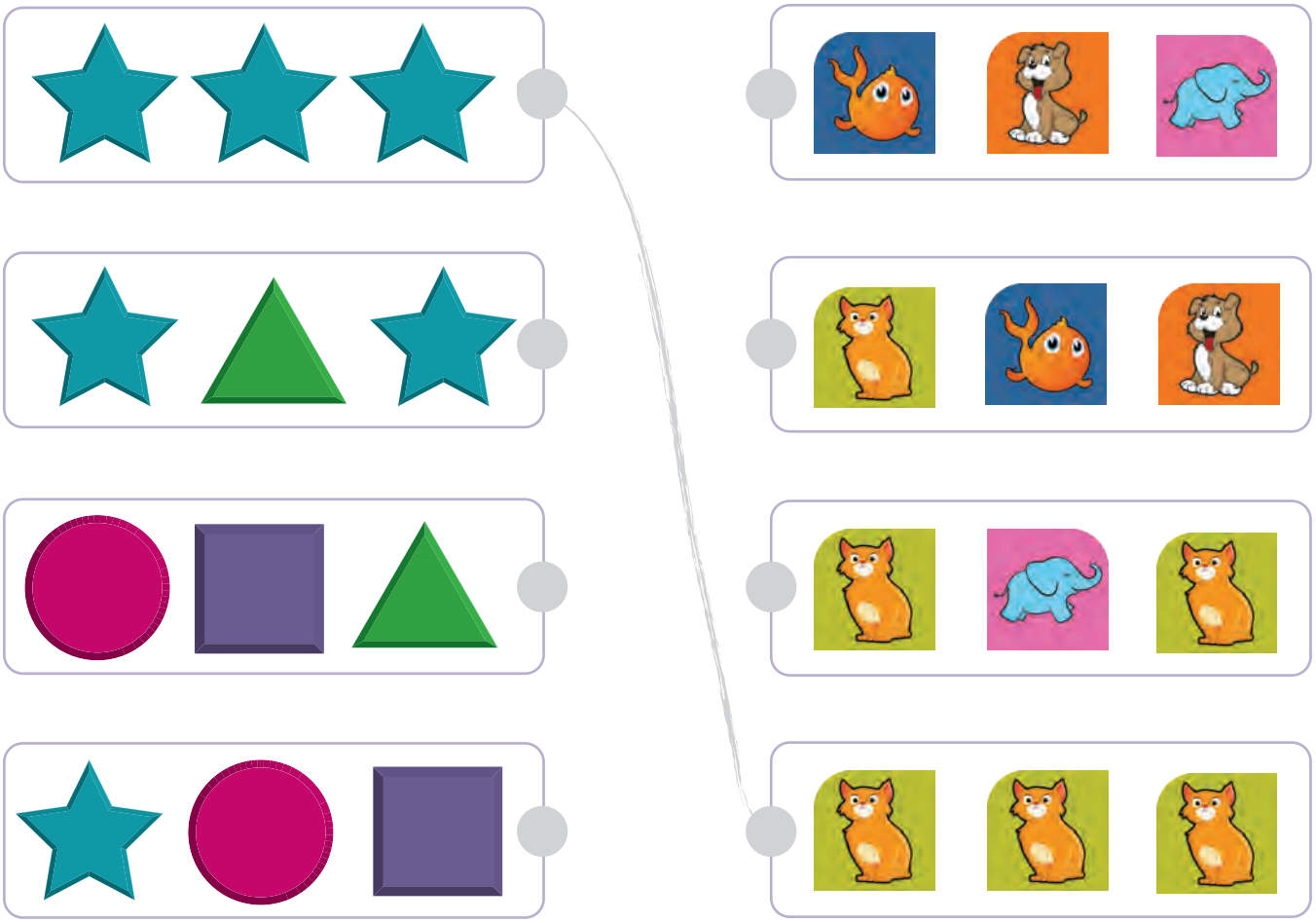
Controlling by Events Assessment

C O
D E

You've been given a magical controller that changes the picture on the frame on your desk.

Take a look below to see what each button does. Can you figure out which series of button events will cause your frame to show the pictures on the right? Draw a line from each set of pictures to the button combination that causes it. The first one has been done for you.





Course C

Lesson 1: Building a Foundation

Unplugged

Overview

New and unsolved problems are often pretty hard. If we want to have any chance of making something creative, useful, and clever, then we need to be willing to attack hard problems even if it means failing a few times before we succeed. In this lesson, students will be building a structure with common materials. The structure will be tested on its ability to hold a textbook for more than ten seconds. Most students will not get this right the first time, but it's important they push through and keep trying.

Purpose

This lesson teaches that failure is not the end of a journey, but a hint for how to succeed. The majority of students will feel frustrated at some point in this lesson, but it's important to emphasize that failure and frustration are common steps to creativity and success.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Try, Try Again**

Main Activity (20 min)

- **Building a Foundation**

Wrap Up (10 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Extended Learning

Objectives

Students will be able to:

- Outline steps to complete a structural engineering challenge.
- Predict and discuss potential issues in structure creation.
- Build a structure based on team plan.
- Revise both the plan and the structure until they satisfy challenge.

Preparation

- [ ] Watch the **Building a Foundation - Teacher Video**.
- [ ] Watch the **Building a Foundation - Lesson in Action Video**.
- [ ] Print **Building a Foundation - Teacher Prep Guide**.
- [ ] Gather enough building elements (marshmallows or gumdrops with toothpicks or popsicle sticks) for each group. You don't have to give any certain amount; just make sure you put some limit on materials.
- [ ] Give a **Think Spot Journal** to each student.

Links

For the Teacher

- **Building a Foundation** - Teacher Video
- **Building a Foundation** - Lesson in Action Video
- **Building a Foundation** - Teacher Prep Guide
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Persistence** - Trying again and again, even when something is very hard.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

Persistence - Say it with me: Per-sis-tence

Trying again and again, even when something is very hard

Try, Try Again

- Does everyone get everything right the first time?
- When I was a baby learning to walk, did I stand up and run off on my first try?
- Sometimes, the best and most useful things to do are the hardest to learn.
  - It can take a while to learn hard things
  - If you don't do something well at first, does it mean that you never will?
  - Can you think of something that was hard at first, but that you can now do pretty easily?
    - Walking
    - Talking
    - Riding a bike
- When you fail at doing something, you get a hint at what went wrong. You just need to look for it.
  - If your bike tips over, next time you need to work on balance.
  - If you're filling a balloon and it pops, next time you need less air.
- Think of the mistakes as chances to learn how to do something better next time.

Lesson Tip

Here are some great resources to prep your class with the concept of persistence before you turn them loose on this project:

- **Mouse Wants a Cracker**
- **Fall 7 Times, Stand Up 8**
- **Never Ever Give Up**
- **If You Quit Too Soon**

Main Activity (20 min)

Building a Foundation

Have you ever started on a task, then discovered that it was much harder than you thought it would be? Hard tasks can make us want to give up, but if we stick to our goal and keep trying, then we just might make something better than we've ever made before!

In this challenge, we'll work to construct towers that are strong enough to hold a textbook for at least 10 seconds, using everyday materials.

Rules:

- Use only the supplies provided to build a tower.
- The tower can be any shape, but it has to be at least as tall as the paper cup.
- The tower must support the weight of a book for a full 10 seconds.

Directions:

1. Divide students into groups of three or four.
2. Explain the rules of the challenge, given above.
3. Provide each group with limited supplies and make it known that they will get no more.
4. Challenge the class to think ahead to the problem and plan out their method of building their first tower.

5. Encourage students to begin building, then have them alert you when they think they've met the challenge described by the rules.
6. Test each structure. Is it taller than the cup? Does it hold a book?
7. If not, have students enter a cycle of planning, fixing, testing, and planning again until the challenge has been met.

Lesson Tip

The planning stage can be difficult for young students. It may be helpful for you to place some idea "examples" at the front of the room. Do not announce that they are there. Simply encourage students to take a walk if they get frustrated. Try to encourage students to locate the tips on their own if at all possible.

8. Congratulate the students as they succeed and take pictures of the successful towers!

Wrap Up (10 min)

Flash Chat: What did we learn?

- Were you proud of what you made?
- Do you think you could make a tower as tall as a chair that could hold a person?
  - How many gumdrops do you think you would need?
- Was there a time that you thought about giving up?
  - How did you get past that feeling?

Lesson Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future. We provide a **Think Spot Journal** as a basic template for students to use as their daily journal.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw a picture of your structure.
- What were some problems you ran into while building? How did you fix these problems?

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Try It Again!

Try doing the same activity with different materials.

Building a Foundation

Learning Persistence through Challenges

Directions:

1) Divide students into groups of 3 or 4.

2) Explain the rules of the challenge, provided on the other page.

3) Provide each group with limited supplies and make it known that they will get no more.

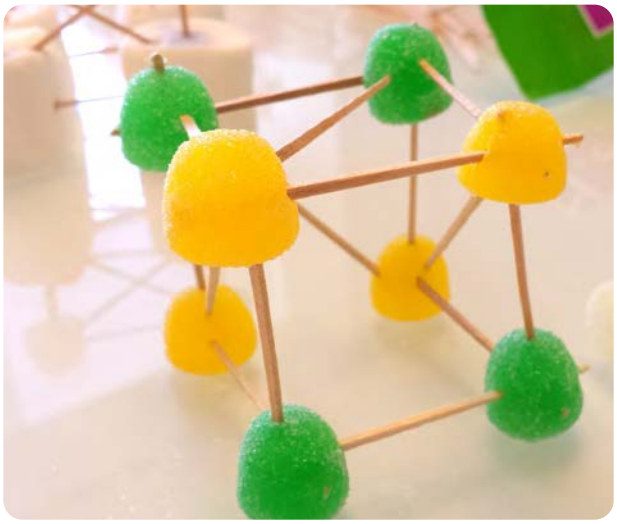
4) Challenge the class to think ahead to the problem and plan out their method of building their first tower.

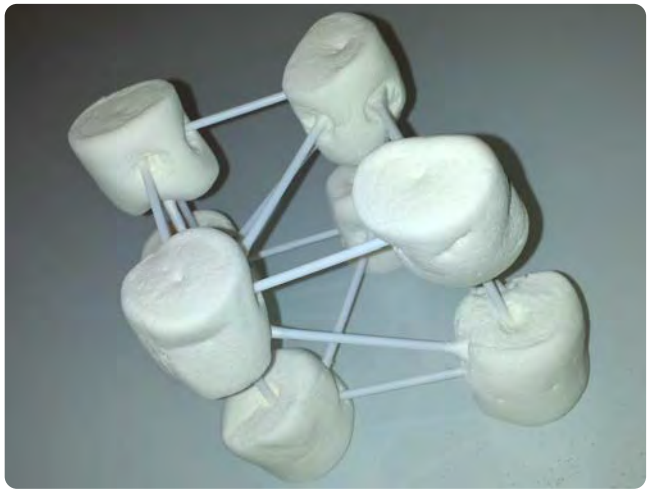
5) Encourage students to begin building, then have them alert you when they think they've met the challenge described by the rules.

6) Test each structure. Is it taller than the cup? Does it hold a book?

7) If not, have students enter a cycle of planning, fixing, testing, and planning again until the challenge has been met.

8) Congratulate the students as they succeed and take pictures of the successful towers (if possible) to upload to the Code.org site!





Lesson 4: Real-Life Algorithms: Paper Airplanes

Unplugged | Algorithms

Overview

In this lesson, students will relate the concept of algorithms back to everyday activities. After discussing algorithms, students will make paper airplanes using an algorithm. The goal here is to start building the skills to translate real world situations to online scenarios and vice versa.

Purpose

In this lesson, students will learn that algorithms are everywhere in our daily lives. For example, there is a specific algorithm to plant a seed. Instead of giving vague or over-generalized instructions, students will break down a large activity into smaller and more specific instructions. From these instructions, students must determine a proper order for the sequence of instructions to be in.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **What We Do Daily**

Main Activity (20 min)

- **Real-Life Algorithms: Paper Airplanes - Worksheet**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (15 min)

- **Daily Algorithms - Assessment**

Extended Learning

Objectives

Students will be able to:

- Decompose large activities into a series of smaller events.
- Arrange sequential events into their logical order.

Preparation

- [ ] Watch the **Real-Life Algorithms: Paper Airplanes - Teacher Video**.
- [ ] Watch the **Real-Life Algorithms: Paper Airplanes - Lesson in Action Video**.
- [ ] Gather paper for students to construct paper airplanes from.
- [ ] Print out **Real-Life Algorithms: Paper Airplanes - Worksheet** for each student.
- [ ] Print **Daily Algorithms - Assessment** for each student.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Real-Life Algorithms: Paper Airplanes** - Teacher Video
- **Real-Life Algorithms: Paper Airplanes** - Lesson in Action Video
- **Real-Life Algorithms: Paper Airplanes** - Worksheet
- **Daily Algorithms** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Algorithm** - A list of steps to finish a task.

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one vocabulary word that is important to review:

Algorithm - Say it with me: Al-go-ri-thm

A list of steps to finish a task.

What We Do Daily

- Ask your students what they did to get ready for school this morning.
  - Write their answers on the board.
  - If possible, put numbers next to their responses to indicate the order that they happen.
    - If students give responses out of order, have them help you put them in some kind of logical order.
    - Point out places where order matters and places where it doesn't.
- Introduce students to the idea that it is possible to create algorithms for the things that we do everyday.
  - Give them a couple of examples, such as making breakfast, brushing teeth, and planting a flower.
- Let's try doing this with a new and fun activity, like making paper airplanes!

Main Activity (20 min)

Real-Life Algorithms: Paper Airplanes - Worksheet

You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to help each other fold a paper airplane.

You know your classroom best. As the teacher, decide if students should do this individually, in pairs, or in small groups.

Directions:

1. Cut out the steps for making a paper airplane from the worksheet provided in the resources.
2. Work together to choose the six correct steps from the nine total options.
3. Glue the six correct steps, in order, onto a separate piece of paper.
4. Trade the finished algorithm with another person or group and let them use it to make their plane!

Lesson Tip

If deciding on the correct steps seems too difficult for your students, do that piece together as a class before you break up into teams.

Lesson Tip

If you are concerned about injury when your students begin flying their paper airplanes, we recommend having them blunt the tip of the plane by either folding it inward or ripping it off and covering the ripped edges with tape.

Wrap Up (15 min)

Flash Chat: What did we learn?

- How many of you were able to follow your classmates' algorithms to make your airplanes?
- Did we leave anything out when making the plane?
  - What would you have added to make the algorithm even better?
  - What if the algorithm had been only one step: "Fold a Paper Airplane"?
    - Would it have been easier or harder?
  - What if it were forty steps?
- What was your favorite part about this activity?

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Can you imagine an algorithm for building a real airplane? What do you think that would look like?
- Write out an algorithm that will take you from your desk to the front of the class.

Assessment (15 min)

Daily Algorithms - Assessment

- Hand out the **Daily Algorithms - Assessment** and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Go Figure

- Break the class up into teams.
- Have each team come up with several steps that they can think of to complete a task.
- Gather teams back together into one big group and have one team share their steps, without letting anyone know what the activity was that they had chosen.
- Allow the rest of the class to try to guess what activity the algorithm is for.

Unplugged

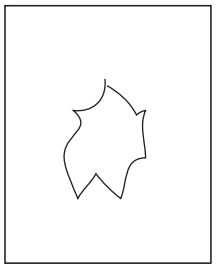
Name: _____ Date: _____

Real-Life Algorithms

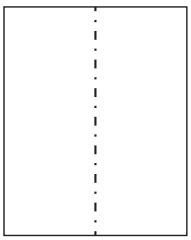
Paper Airplane Worksheet

You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to help each other make paper airplanes.

Cut out the steps of making an airplane below. Glue the six the correct steps, in order, onto a separate piece of paper. Trade your finished algorithm with another person or group and let them use it to make an actual flying model paper plane!

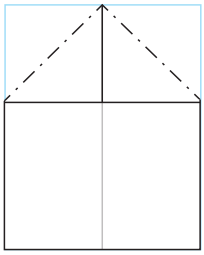


CUT CENTER OUT OF PAPER

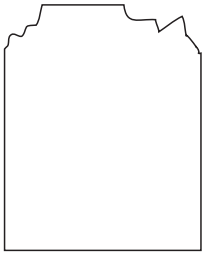


CREASE PAPER DOWN THE CENTER

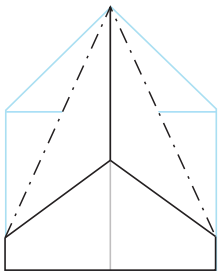
CRUMBLE PAPER



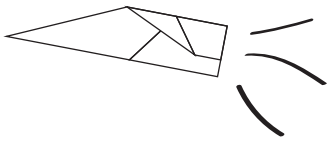
FOLD TOP CORNERS TO CENTER



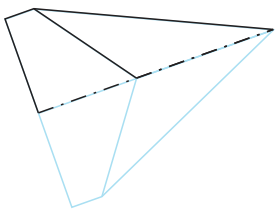
RIP CORNER OFF PAPER



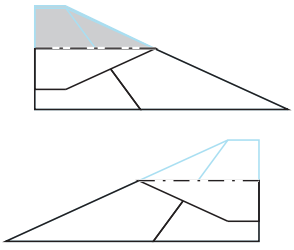
FOLD CORNER SIDES TO CENTER



TOSS FINISHED PLANE



FOLD PAPER IN HALF AGAIN



PULL SIDES DOWN

U

Unplugged

Name: _____ Date: _____

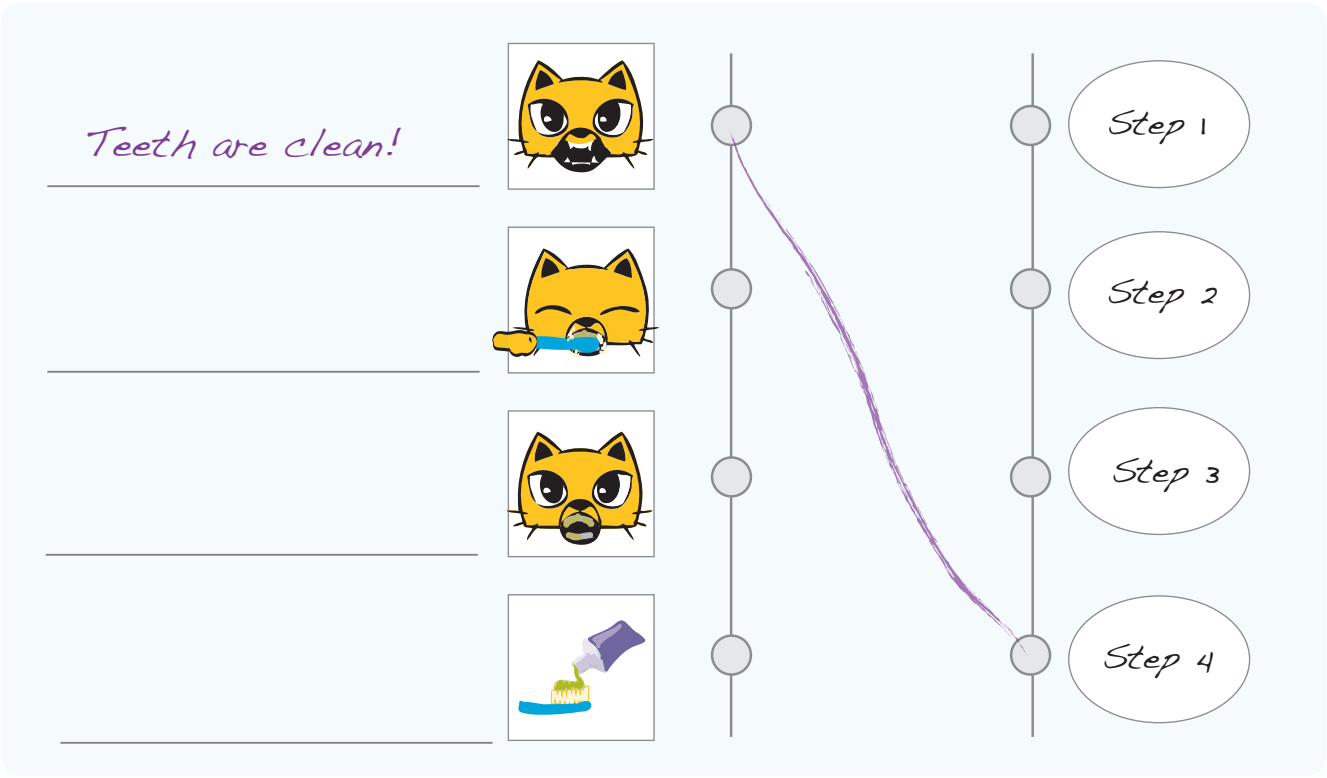
Daily Algorithms

Assessment Worksheet

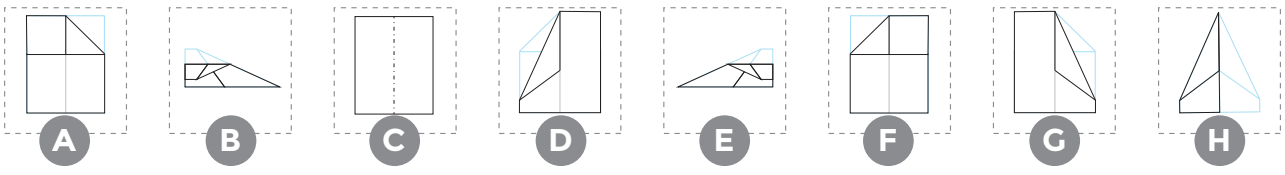
C O D E

An algorithm is a list of instructions for accomplishing a task. We follow algorithms everyday when it comes to activities like making the bed, making breakfast, or even getting dressed in the morning.

These images are not in order. First, describe what is happening in each picture on the line to its left, then match the action to its order in the algorithm. The first one has been done for you as an example.



Sometimes you can have more than one algorithm for the same activity. The order of some of these steps can be changed without changing the final product. Use the letters on the images below to create two algorithms for making a paper airplane.



ALGORITHM 1: _____ _____ _____ _____ _____ _____ _____ _____

ALGORITHM 2: _____ _____ _____ _____ _____ _____ _____ _____

Lesson 7: Getting Loopy

Unplugged | Loops

Overview

Loops are a handy way to repeat actions a certain number of times. In this lesson, students will dance their way to a better understanding of how to use repeat loops.

Purpose

Loops allow for students to categorize their code into what needs to be repeated and what does not. Students will develop critical thinking skills by noticing repetition in movements and determining how many times the to repeat commands to develop those loops.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **Repeat After Me**

Main Activity (15 min)

- **Getting Loopy - Worksheet**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (10 min)

- **Getting Loopy - Assessment**

Extended Learning

Objectives

Students will be able to:

- Repeat actions initiated by the instructor.
- Translate a picture program into a live-action dance.
- Convert a series of multiple actions into a single loop.

Preparation

- [ ] Watch the **Getting Loopy - Teacher Video**.
- [ ] Print one **Getting Loopy - Worksheet** for the class.
- [ ] Print one **Getting Loopy - Assessment** for each student.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Getting Loopy** - Teacher Video
- **Getting Loopy** - Worksheet
- **Getting Loopy** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important vocabulary word:

Loop - Say it with me: Loop

The action of doing something over and over again

Repeat After Me

- Ask for a volunteer and have them stand.
  - Instruct your volunteer to walk around the table (or their chair, or a friend).
  - When they finish, instruct them to do it again, using the exact same words you did before.
  - When they finish, instruct again.
  - Then again.
- Would it have been easier for me to just ask you to go around the table four times?
  - What if I wanted you to do it ten times?
- If I want you to repeat an action ten times in a row, that's called "looping."
- When I know in advance that I want you to do something a certain number of times, it's easier for both of us if I just ask you to "Repeat it that many times."
- Can you think of some other things that we could loop?

Main Activity (15 min)

Getting Loopy - Worksheet

Today, we're going to have a dance party!

Sometimes, when you know that you will be doing something over and over, it is helpful to know how many times it needs to be done before you begin. That way, you can keep track of how many actions you have left as you go.

Example:

If your mom wanted you to play her favorite song over and over, she wouldn't say:

"Please play my song, play my song, play my song, play my song."

She would most likely say:

"Please play my song four times."

Directions:

- Look at the dance moves provided on the **Resource not found**.

Lesson Tip:

Looking for some good music? Here are some great places to find some:

- **Radio Disney**
- **Nick Radio**
- **Kidz Bop Radio**

Please be advised that some of these stations may display ads with third-party content. If you find that displayed ads are inappropriate, you may want to direct students to a different site, or research ad-blockers that can prevent this content.

- Show the class what the entire dance looks like done at full-speed.
- Run through the dance slowly, one instruction at a time, with the class.
- Can you find the loop in the instructions?
  - What would the dance look like if we only repeated the main part 2 times?
  - What if we repeated the main part 4 times?
- Can you find anything else in the dance that we could use a loop for?

Wrap Up (15 min)

Flash Chat: What did we learn?

- Do you think it is easier to add more pictures to the screen or change the number of times we loop?
  - Would your answer be the same if we wanted to loop 100 times?
- Could we use these same loops with different dance moves?
- Do you know any dances that are done inside a loop?
- What was your favorite part about that activity?

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw a picture of you dancing today. Draw the moves you did in loops, like clapping three times. Write the number of times you repeated that loop.
- What else can you use a loop for?

Assessment (10 min)

Getting Loopy - Assessment

- Hand out the assessment to each student. Allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

So Moving

- Give the students pictures of actions or dance moves that they can do.
- Have students arrange moves and add loops to choreograph their own dance.
- Share the dances with the rest of the class.

Connect It Back

- Find some YouTube videos of popular dances that repeat themselves.
- Can your class find the loops?
- Try the same thing with songs!

The Iteration

Repeat this part 3 times!


Clap


Clap


Clap


Behind Head


Waist


Behind Head


Waist


Clap


Clap


Clap


Left Up


Right Up


Left Up


Right Up


Clap


Clap


Clap

Then do this


Belly Laugh

U

Unplugged

Name: _____ Date: _____

Getting Loopy

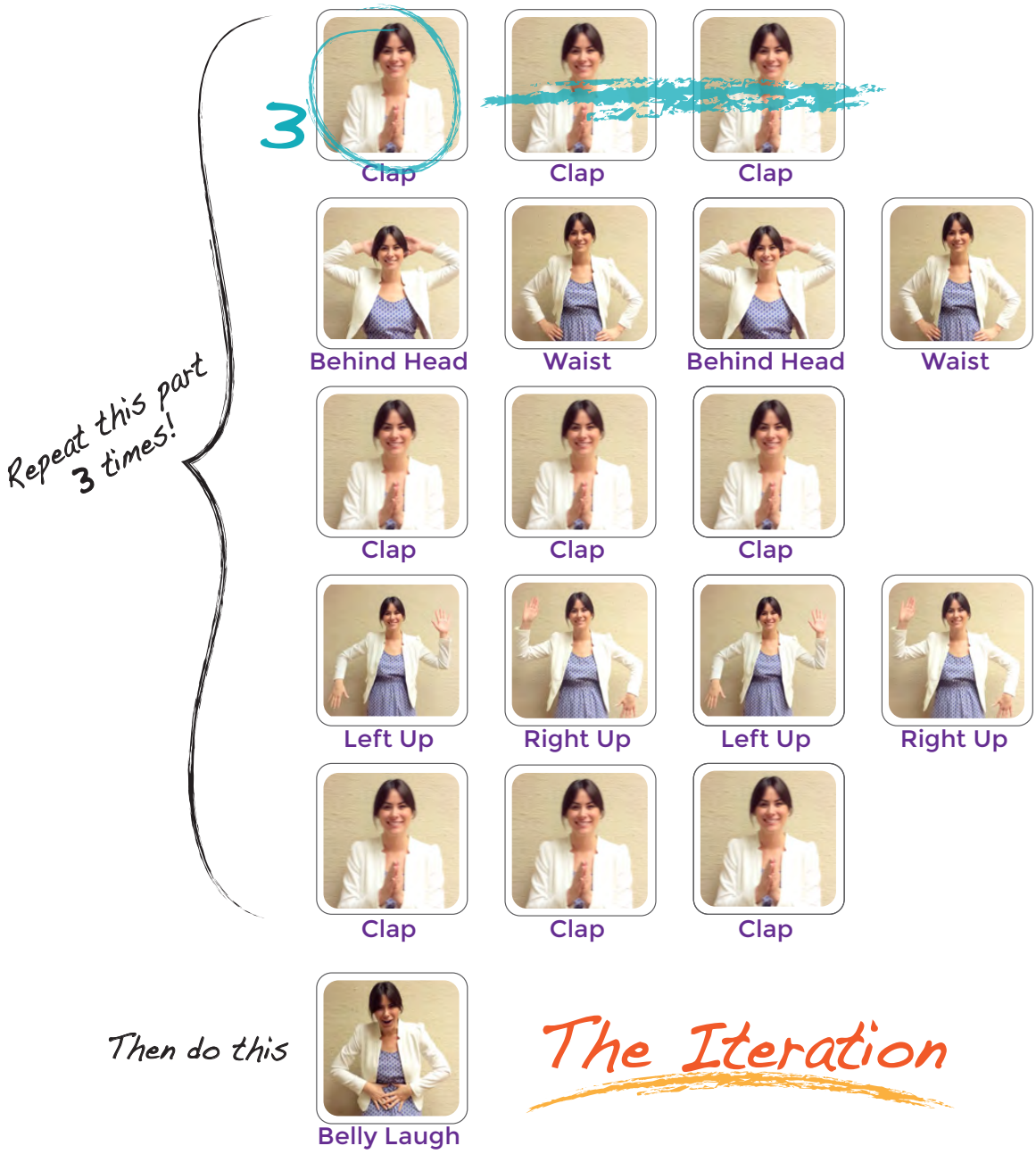
Unplugged Loops Activity

C O D E

Looping can save space!

What if we wanted to take The Iteration dance below and make more loops inside? Can you circle the actions that we can group into a loop and cross out the ones that we don't need anymore? Write a number next to each circle to let us know how many times to repeat the action.

The first line has been done for you.



Lesson 11: Events Unplugged: The Big Event

Unplugged | Events

Overview

Students will soon learn events are a great way to add flexibility to a pre-written algorithm. Sometimes you want your program to be able to respond to the user exactly when the user wants it to. Events can make your program more interesting and interactive.

Purpose

Today, students will learn to distinguish events and actions. The students will see activities interrupted by having a 'button' pressed on a paper remote. When seeing this *event*, the class will react with a unique action. Events are widely used in programming and should be easily recognizable after this lesson.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **A Series of Events**

Main Activity (15 min)

- **CSF The Big Event Activity - Worksheet**

Wrap Up (10 min)

- **Flash Chat: What did we learn?**

Assessment (10 min)

- **CSF The Big Event Activity - Assessment**

Extended Learning

Objectives

Students will be able to:

- Repeat commands given by an instructor.
- Recognize movements of the teacher as signals to initiate commands.
- Practice differentiating pre-defined actions and event-driven ones.

Preparation

- [ ] Watch the **The Big Event - Teacher Video**.
- [ ] Print one **CSF The Big Event Activity - Worksheet** and Event Controller.
- [ ] Print one **CSF The Big Event Activity - Assessment** for each student.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **CSF The Big Event Activity** - Worksheet
- **The Big Event** - Teacher Video
- **CSF The Big Event Activity** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Event** - An action that causes something to happen.

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important vocabulary word:

Event - Say it with me: E-vent

An event is an action that causes something to happen.

A Series of Events

- Prep your class to answer a question:
  - "I'm going to ask you a question. I want you to raise your hand if you want me to call on you for the answer."
  - Ask a simple question that most of your students should be able to answer, such as:
    - How many thumbs do I have?
    - What is bigger, a bird or a horse?
  - Call on a student who has their hand raised and let them give their answer.
  - Upon finishing that display, ask the class how you knew that the student wanted you to call on them.
    - Your class will likely mention the raising of the hand.
  - Explain to everyone that when students raise their hand, it is an "event" that causes you to know that they want to be called on.
- Ask the class if they can think of any other events that give signals.
  - You may need to remind them that you're not talking about an event like a birthday party or a field trip.
  - If they have trouble, you can remind them that an event is an action that causes something to happen.
    - What about an alarm clock going off? What does that make happen?
    - What about pressing "Start" on the microwave? What does that do?
    - What about pressing the power button on your tv remote?
- Today, we're going to practice changing programs by introducing events.

Main Activity (15 min)

CSF The Big Event Activity - Worksheet

- Do you remember guiding Red from Angry Birds to the pig in the Maze puzzles?
  - In that exercise, you knew in advance exactly where you wanted Red to go, so you could make a program that took Red from start to finish without any interruptions.
  - In most real programs, we can't do that because we want to have options, depending on what the user needs.
    - Say that I only want my character to move when my finger is on the screen of my phone. I would need to program the character to only move when I put my finger on the screen of my phone.
    - Putting my finger on the screen would then become an "event" that tells my character to move.

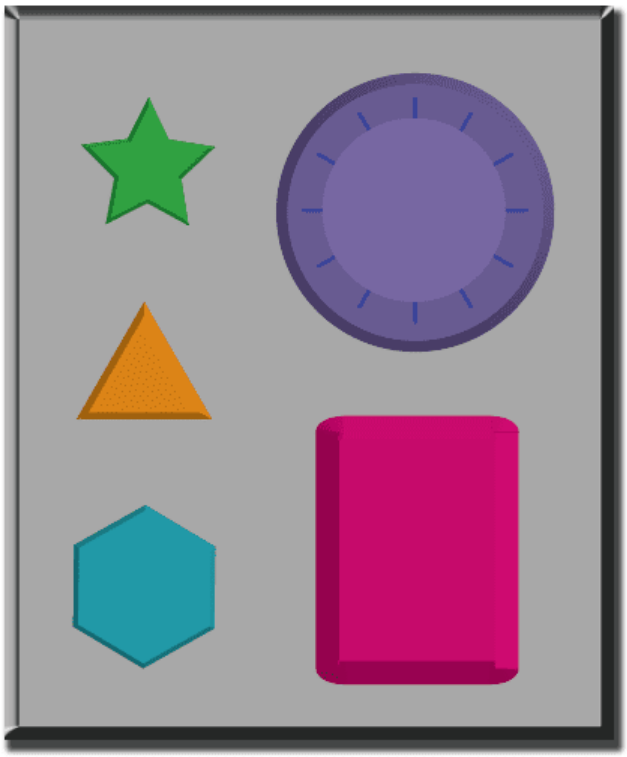
Lesson Tip

If your students seem confused, talk about their favorite games and all of the ways that they let the characters know what they're supposed to do. Point out how the game would be really boring if it ran from start to finish without any events required.

In earlier lessons, we created algorithms that allowed us to control a friend or other character for several steps at a time. It was fun and useful, but what happens when you don’t know everything that you want your friend to do in advance? This is where events come in!

Directions:

- Project the Event Controller onto your classroom screen.



- Decide with your class what each button does. We suggest:
  - Pink Button -> Say “Wooooo!”
  - Teal Button -> “Yeah!”
  - Purple Dial -> “Boom!”
  - Green Button -> Clap
  - Orange Dial -> Stomp
- Practice tapping the buttons on the overhead and having your class react.
- Add some button sequences into the mix and have the students try to keep up with their sounds.
- Let your class know that every time you push a button, it is an “event” that lets them know what they are expected to do next.
- Get the class started on a planned task before interrupting them again with the buttons. We suggest:
  - Counting to 10
  - Singing “Old MacDonald”
- Once their plan is underway, interject button presses sporadically.
- Continue the blend until they understand the difference between actions that are guided by a plan and those that are event driven.

Wrap Up (10 min)

Flash Chat: What did we learn?

- Why do we need to be able to handle events in a program?
- What are some other kinds of events that you can think of?

Assessment (10 min)

CSF The Big Event Activity - Assessment

- Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

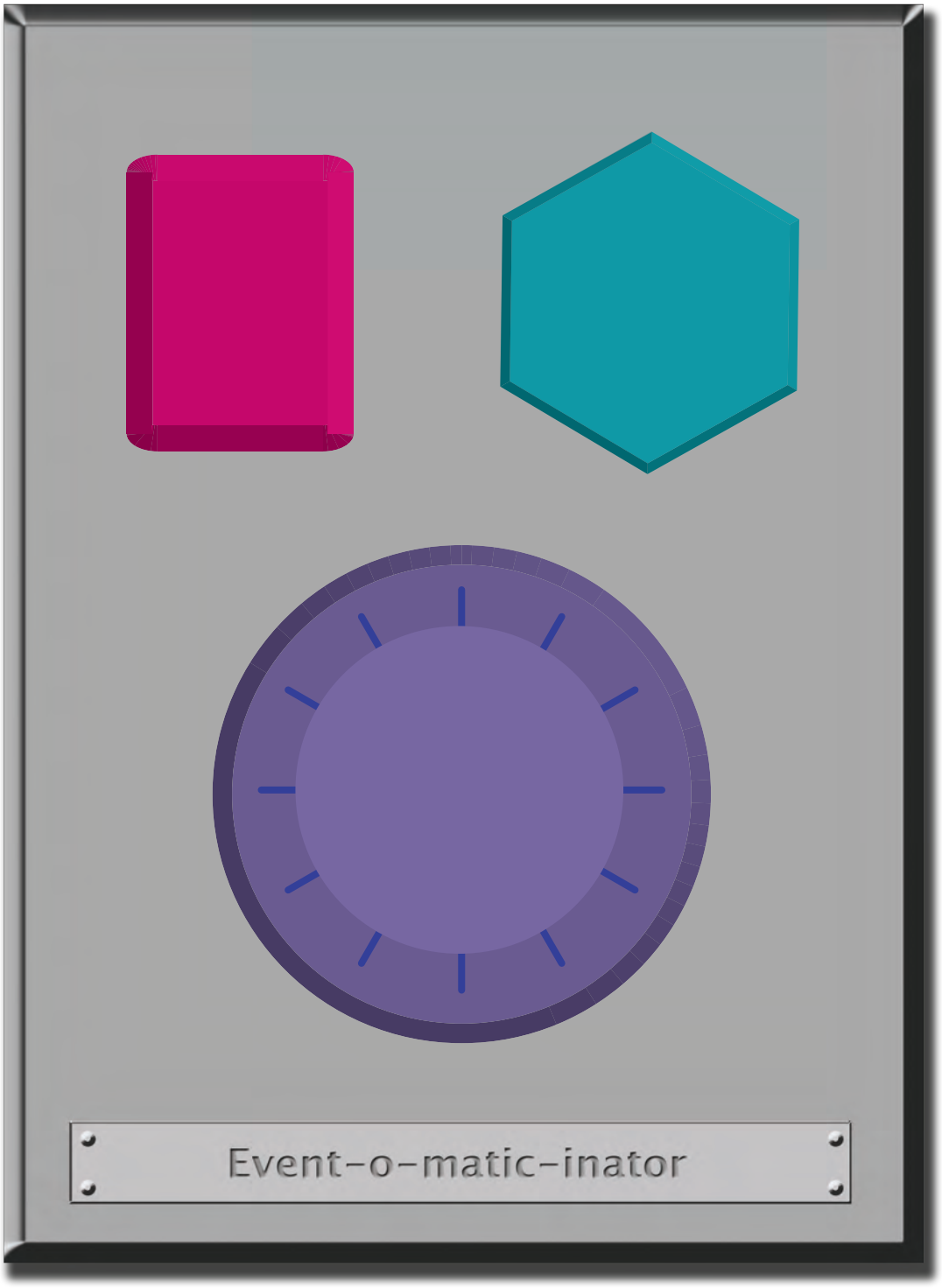
Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

One Person's Event is Another One's Reaction

- Assign each student an event to watch out for, and an appropriate reaction to that event. Chain the actions so that each child's reaction becomes an event that triggers the reaction of another student. Keep assigning until everyone has something to do and everyone makes someone react.

Eventopalooza

- Break the class up into groups. Using the Events Controller, assign each group a different reaction to the same button. Do this for all three buttons, then watch the chaos!


Event-o-matic-inator

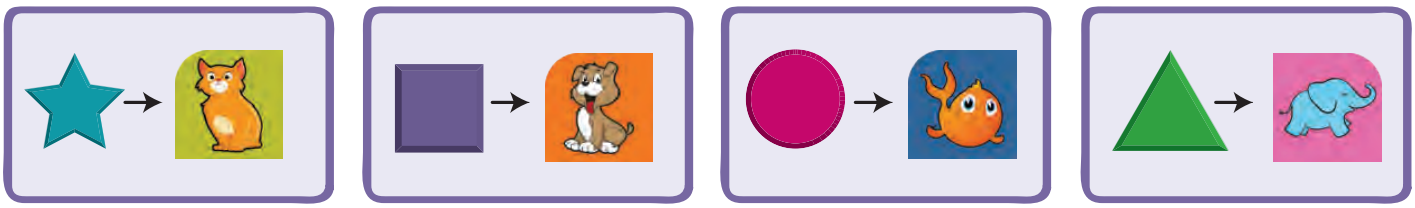
Name: _____ Date: _____

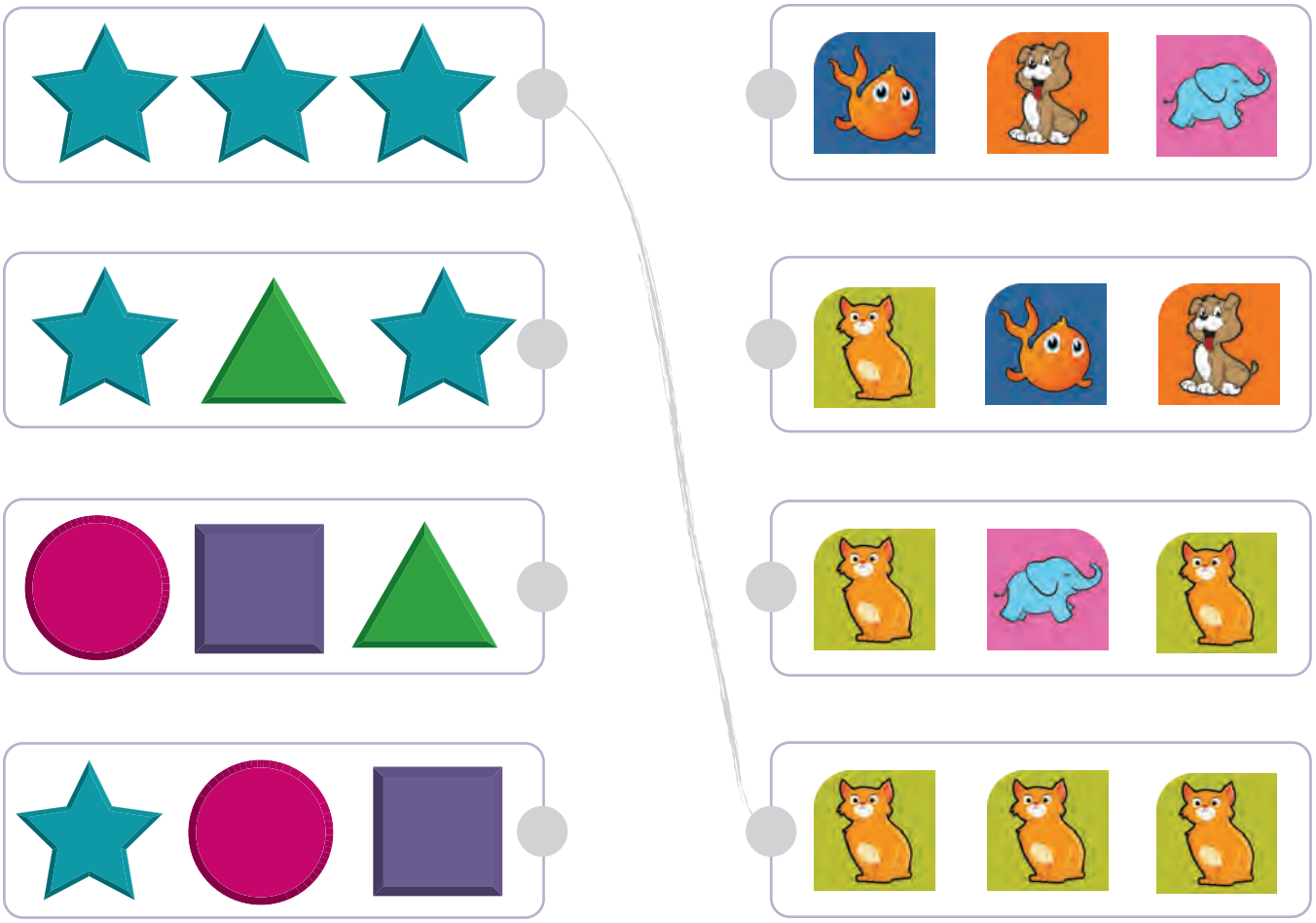
The Big Event

Controlling by Events Assessment

You've been given a magical controller that changes the picture on the frame on your desk.

Take a look below to see what each button does. Can you figure out which series of button events will cause your frame to show the pictures on the right? Draw a line from each set of pictures to the button combination that causes it. The first one has been done for you.





Lesson 14: Common Sense Education: Screen Out the Mean

Common Sense Education | Cyberbullying | Unplugged

Overview

This lesson helps children to recognize that it is essential to tell a trusted adult if something online makes them feel angry, sad, or scared.

Students learn that other people sometimes can act like bullies when they are online. They will explore what cyberbullying means and what they can do when they encounter it. After reading a scenario about mean online behavior, students discuss what cyberbullying is, how it can make people feel, and how to respond. Finally they use their knowledge to create a simple tip sheet on cyberbullying in their **Think Spot Journal**.

Purpose

Students may not ever have the misfortune of experiencing cyberbullying, but we want to make sure that the students are prepared for and knowledgeable about it, in case they ever witness it during an online situation. Students will learn how to identify cyberbullying and what steps they should take to make it stop. This may become helpful in later puzzles when students have the opportunity to share their work. If someone negatively responds to a student's work, this lesson will provide them with the tools that they need to handle the situation.

Agenda

Warm Up (5 min)

- **Introduction**

Main Activity (35 min)

- **What Is Cyberbullying?**
- **What to Do About Cyberbullying**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (5 - 10 min)

- **Screen Out the Mean - Teacher Prep Guide**

Objectives

Students will be able to:

- Analyze online behaviors that could be considered cyberbullying.
- Explain how to deal with a cyberbullying situation.
- Recognize the importance of engaging a trusted adult if the student experienced cyberbullying.

Preparation

- [ ] Review **Screen Out the Mean - Teacher Prep Guide** from Common Sense Education's website.
- [ ] Print out a worksheet from the link above (page 6) for each student.
- [ ] Print out an assessment from the link at the top (page 7) for each student.
- [ ] Make sure every student has a **Think Spot Journal**.
- [ ] Print or display the **Online Safety Poster - Student Handout** for the class to see.

Links

For the Teacher

- **Screen Out the Mean** - Teacher Prep Guide
- **Common Sense Education** - Website
- **Think Spot Journal** (**PDF** | **DOCX**)

For the Students

- **Online Safety Poster** - Student Handout

Vocabulary

- **Cyberbullying** - Doing something on the internet, usually again and again, to make another personal feel angry, sad, or scared.
- **Online** - Connected to the internet.

Teaching Guide

Warm Up (5 min)

Introduction

Encourage:

Encourage students to share what they know about bullying.

Ask:

- What kinds of things count as bullying?
  - Students should understand that bullying is behavior that is purposefully mean or scary to someone else. For example, making fun of how someone looks, telling lies about them, or threatening to do something bad to them.
- How does bullying make other people feel?
  - Hurt, angry, upset, scared
- What is the best thing to do when you feel bullied, or when you see someone else being bullied?
  - Students should know to always tell a trusted adult when they experience or witness bullying.

Explain:

Students will be learning about a kind of bullying that can take place when they use the internet.

Main Activity (35 min)

common sense education®

What Is Cyberbullying?

Define:

- *Online*: Connected to the internet
- *Cyberbullying*: Doing something on the internet, usually again and again, to make another person feel angry, sad, or scared

Discuss:

Some kids do not go online very much at all, either because of their family's rules or because they do not like it very much. Other kids do go online to do different things.

Ask:

- What do you do online, or what do you think you might like to do?
  - Students may mention activities like sending messages to friends and playing games.

Share:

Most of the time when students go online it is to do fun or interesting things. But sometimes people can be mean to each other online and this is called cyberbullying.

Ask:

- Did you ever see someone make someone else feel bad online?
  - Answers will vary. Remind students to tell what happened, but not to use real names.

Explain:

Tell students that they will be learning more about how cyberbullying occurs, and what to do when it happens to them or to someone they know.

What to Do About Cyberbullying

Discuss:

Read aloud these two scenarios and discuss them briefly with the class.

- Kyle keeps getting instant messages from someone saying means things about him. The person who is sending the messages doesn't use a real name, but Kyle can tell the messages are coming from someone who also makes fun of him at school in gym class.
- Sasha is a new girl at school, and she's making a lot of friends. Then Sasha finds out that another girl sent around an email that had a picture of a cow with Sasha's name on it.

Next, pass out the **Screen Out the Mean - Teacher Prep Guide** worksheet from page 6. Read aloud the story at the top and ask students to work in pairs or groups to finish the worksheet.

Ask the class to discuss to Jada's story. Tell the class there are specific steps to handling a cyberbully.

Discussion Goal

Questions to stimulate discussion include:

- What do you think happened to Jada's game?
- How do you think Jada, Kyle, or Sasha felt when these things happened to them?
- How do you know if someone is cyberbullying you?
- Why do you think it is important to stop using the computer when someone starts cyberbullying you?
  - It's possible that if students stay online, the cyberbullying may continue or get worse.

Teacher Tip:

These scenarios can be read all at once and discussed as a whole, or be read and discussed individually.

- Jada should STOP using the computer.
- Jada should TELL an adult she trusts what happened.
- Jada should not go back online or return to the pony website until an adult says it is OK.
- If Jada and Michael are good friends, Jada may want to tell Michael how his actions made her feel after she gets help from an adult.
- If Michael continues cyberbullying her, she should play with other kids who don't cyberbully others.

In general, there are four steps students should take if they or someone they know are experiencing cyberbullying.

1. Stop using the computer until it is safe.
2. Tell an adult you trust.
3. Go online only when a trusted adult says it is okay.
4. Play online only with kids who you know and are nice.

Wrap Up (15 min)

Flash Chat: What did we learn?

Ask:

- What is cyberbullying. How does it make people feel?
  - Students should recognize that cyberbullying is any kind of online behavior that makes people feel sad, scared, angry or upset.
- What four things can you do to help stop cyberbullying?

  S. *Stop* using the computer until it is safe.

  T. *Tell* an adult you trust.

  O. Go *Online* only when a trusted adult says it is okay.

  P. *Play* online only with kids who are nice.
- What is the most important thing to do if someone starts cyberbullying you?
  - Telling a trusted adult is the most important response whenever someone makes them feel sad, scared, or angry online.

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Write down the names of some trusted adults you can go to if you ever feel bullied.
- What are the four steps you should take if you or someone you know is being cyberbullied.

Assessment (5 - 10 min)

Screen Out the Mean - Teacher Prep Guide

Pass out an assessment to each student. Allow students a few minutes to complete it then review the answers (page 9 of the link above) with the class. If there's time, allow for a discussion about the questions.

Screen Out the Mean

Essential Question

What can you do when someone is mean to you online?

Lesson Overview

Students learn that children sometimes can act like bullies when they are online. They explore what cyberbullying means and what they can do when they encounter it.

Students first read a scenario about mean online behavior. They then discuss what cyberbullying is, how it can make people feel, and how to respond. Then they use their knowledge to create a simple tip sheet on cyberbullying. Students recognize that it is essential to tell a trusted adult if something online makes them feel angry, sad, or scared.

Learning Objectives

Students will be able to ...

- analyze online behaviors that could be considered cyberbullying.
- explain how to deal with a cyberbullying situation.
- recognize the importance of engaging a trusted adult when they experience cyberbullying.

Materials and Preparation

- Copy the **STOP Cyberbullying Student Handout**, one for each student.
- Preview the scenario in Teach 2 and be prepared to present it to the class.

Family Resources

- Send home the **Cyberbullying Family Tip Sheet (Elementary School)**.

Estimated time: 45 minutes

Standards Alignment –

Common Core:

grade K: RL.1, RL.2, RL.3, RL.4, RL.10, RI.1, RI.2, RI.3, RI.4, RI.10, RF.4, W.2, W.5, W.7, W.8, SL.1a, SL.1b, SL.2, SL.3, SL.4, SL.5, SL.6, L.6

grade 1: RL.1, RL.2, RL.3, RL.4, RI.1, RI.2, RI.3, RI.4, RI.10, RF.4a, W.5, W.7, W.8, L.6

grade 2: RL.1, RL.2, RL.3, RI.4, RI.6, RI.10, W.2, W.7, W.8, RF.4a, SL.1a, SL.1b, SL.1c, SL.2, SL.3, L.6

NETS·S: 2a, 5a, 5d

Key Vocabulary –

online: connected to the Internet

cyberbullying: doing something on the Internet, usually again and again, to make another person feel angry, sad, or scared

introduction

Warm-up *(5 minutes)*

ENCOURAGE students to share what they know about bullying.

ASK:

What kinds of things count as bullying?	Students should understand that bullying is behavior that is purposely mean or scary to someone else – for example, making fun of how someone looks, telling lies about them behind their back, or threatening to do something bad to them.
How does bullying make other people feel?	Sample responses: • Hurt • Angry • Upset • Scared
What is the best thing to do when you feel bullied, or when you see someone else being bullied?	Students should know to always tell a trusted adult when they experience or witness bullying.

EXPLAIN to students that they will be learning about a kind of bullying that can take place when they use the Internet.

teach 1

What Is Cyberbullying? *(15 minutes)*

DEFINE the Key Vocabulary term **online.**

DISCUSS the fact that some kids don't go online very much at all, either because of their family's rules or because they don't like it very much. Other kids do go online to do different things.

ASK:

What do you do online, or what do you think you might like to do?	Students may mention sending emails, instant messaging, and playing games.

SHARE with students that most of the time when they go online it is to do fun or interesting things. But sometimes people can be mean to each other online and this is called cyberbullying.

DEFINE the Key Vocabulary term **cyberbullying.**

EMPHASIZE that when children are mean to someone else online, even if they only do it one time, it isn't nice. Also stress that cyberbullies usually bully repeatedly, with the intention of causing hurt feelings. When children do something very mean and/or scary, or do it over and over again, then they are cyberbullying.

SHARE with students some examples of **cyberbullying**. These might include:

- sending a mean email or IM to someone
- posting mean things about someone on a website
- making fun of someone in an online chat
- doing mean things to someone's character in an online world like Club Penguin or WebKinz

ASK:

Did you ever see someone make someone else feel bad online?	Answers will vary. Reminds students to tell what happened, but not use real names.

EXPLAIN to students that they will be learning more about how cyberbullying occurs, and what to do when it happens to them or to someone they know.

teach 2

What to Do About Cyberbullying *(20 minutes)*

DISTRIBUTE the **STOP Cyberbullying Student Handout**, one for every student.

GUIDE students through the scenario on the handout. After allowing students time to read it on their own, you may wish to read it aloud.

> *Jada's parents let her play on a website where she can take care of a pet pony and decorate its stall. Her friend Michael has played with her in the past and knows her user name and password. One day Jada goes to the site to care for her pony. She finds that her pony's stall is a mess and that there are some things missing.*

ENCOURAGE the class to answer the questions on their handouts. Invite them to share their answers.

ASK:

What do you think happened?	Students should conclude that Michael went to the website himself and messed up the pony's stall.
How do you think this made Jada feel?	Students should recognize that Michael's behavior probably made Jada feel upset, sad, angry, or let down by her friend.

DIRECT students' attention to the four rules for dealing with cyberbullying at the bottom of their **STOP Cyberbullying Student Handout**. Use the following questions to guide discussion.

ASK:

How will you know when someone is cyberbullying you?	Students should recognize that they may be experiencing cyberbullying whenever someone does something online that makes them feel sad, scared, angry, or upset in any way.
Why do you think it is important to stop using the computer when someone starts cyberbullying you?	Students should realize that if they stay online, the cyberbullying may continue or get worse.

If someone makes you feel angry, sad, or scared online, which grown-ups can you tell and ask for help?	Students may name parents or grandparents, an older sister or brother, a teacher, or the school nurse or counselor. If students cannot think of someone right away, help them brainstorm and identify an appropriate adult.
Why is it important to go online only with an adult, or when an adult says it is OK?	Students should recognize that adults can help guide them online and keep them safe from cyberbullying.
How can you decide whether you should play or chat with someone online?	Students should acknowledge that they need adult guidance in deciding who to connect with online. If someone is very mean to them, or is mean repeatedly, then that person is a cyberbully and should not be contacted online. Remind students that they should never talk to strangers online either without asking a trusted adult, even if that person is nice or has shared interests.
Which of the four things do you think is the most important?	Students should recognize that telling an adult is the single most important thing they should do if they experience or witness cyberbullying.

REVISIT the scenario in the **STOP Cyberbullying Student Handout**, and have students apply the S-T-O-P rules to Jada's situation.

- Jada should STOP using the computer.
- Jada should TELL an adult she trusts what happened.
- Jada should not go back online or return to the pony website when an adult says it is OK.
- If Jada and Michael are good friends, Jada may want to tell Michael how his actions made her feel, after she gets advice from an adult.
- But if Michael continues cyberbullying her, she should play with other kids who don't take part in cyberbullying.

closing

Wrap-up *(5 minutes)*

You can use these questions to assess your students' understanding of the lesson objectives.

ASK:

What is cyberbullying? How does it make people feel?	Students should recognize that cyberbullying is any kind of online behavior that makes people feel sad, scared, angry, or upset.
What four things can you do to help stop cyberbullying?	Students should be able to explain each of the four rules on the **STOP Cyberbullying Student Handout**.
What is the most important thing to do if someone starts cyberbullying you?	Students should understand that telling a trusted adult is the most important response whenever someone makes them feel sad, scared, or angry online.

Screen Out the Mean

Directions

Jada's parents let her play on a website where she can take care of a pet pony and decorate its stall. Her friend Michael has played with her in the past and knows her user name and password.

One day Jada goes to the site to care for her pony. She finds that her pony's stall is a mess and that there are some things missing.

What do you think happened?

How do you think Jada feels?

What should you do if someone starts cyberbullying you?

STOP using the computer until it is safe.

TELL an adult you trust.

Go ONLINE only when a trusted adult says it's OK.

PLAY online only with kids who are nice.

Screen Out the Mean

1. Draw lines to show which things a cyberbully would do most and which things an in-person bully would do most.

Threatens to pull your hair

Takes your stuff in an online game

Sends mean emails

Hits you

2. A cyberbully might:

a) Write an email to make someone feel scared

b) Say mean things at recess

c) Share a knock-knock joke online

3. What should you do if you are cyberbullied?

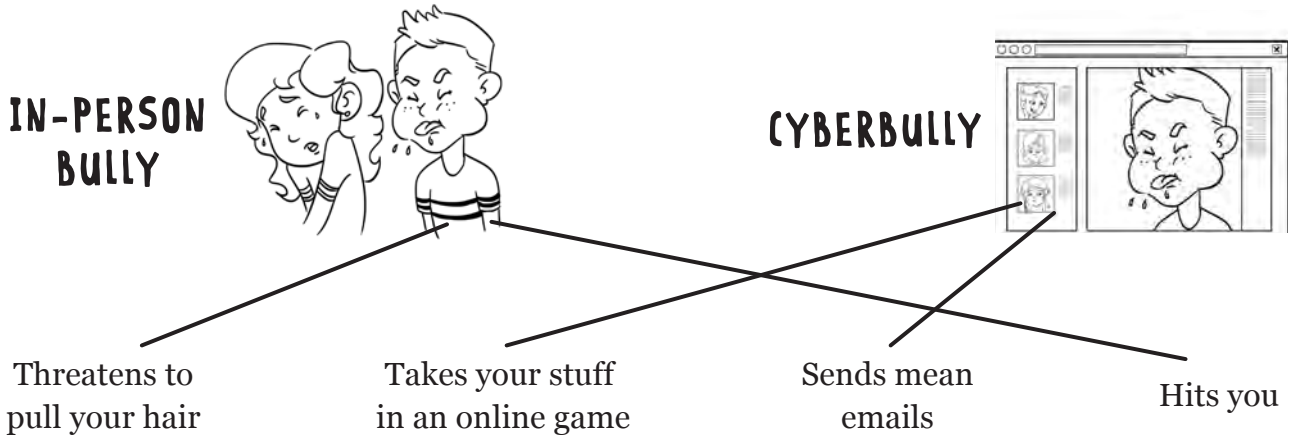
a) Stop using the computer until it is safe

b) Tell an adult you trust

c) Both a and b

Screen Out the Mean

1. Draw lines to show which things a cyberbully would do most and which things an in-person bully would do most.



Answer feedback
A cyberbully does and says mean things online. An in-person bully is mean in person.

2. A cyberbully might:

a) Write an email to make someone feel scared
b) Say mean things at recess
c) Share a knock-knock joke online

Answer feedback
The correct answer is **a**. Cyberbullies are mean online.

3. What should you do if you are cyberbullied?

a) Stop using the computer until it is safe
b) Tell an adult you trust
c) Both a and b

Answer feedback
The correct answer is **c**. If someone is mean to you online, get off the computer and tell an adult. Saying mean things to a cyberbully won't help.

Online Safety

Stop using the computer until it is safe.

Tell an adult you trust.

Go **Online** when a trusted adult says it's OK.

Play online only with kids you know who are nice.

Using Materials From common sense

Lesson 15: Binary Bracelets

Unplugged | Binary

Overview

Binary is extremely important in the world of computers. The majority of computers today store all sorts of information in binary form. This lesson helps demonstrate how it is possible to take something from real life and translate it into a series of ons and offs.

Purpose

In this lesson students will learn how information is represented in a way such that a computer can interpret and store it. When learning *binary*, students will have the opportunity to write codes and share them with peers as secret messages. This can then be related back to how computers read a program, translate it to binary, use the information in some way, then reply back in a way humans can understand. For example, when we type a sentence into a document then press save, a computer translates the sentence into binary, stores the information, then posts a message indicating the document has been saved.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **Off and On**

Main Activity (20 min)

- **Binary Bracelets - Worksheet**

Wrap Up (5 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (15 min)

- **Binary Bracelets - Assessment**

Extended Learning

Objectives

Students will be able to:

- Encode letters into binary.
- Decode binary back to letters.
- Relate the idea of storing letters on paper to the idea of storing information in a computer.

Preparation

- [ ] Watch the **Binary Bracelets - Teacher Video**.
- [ ] Watch the **Binary Bracelets - Lesson in Action Video**.
- [ ] Gather markers for the bracelets. Other decorations like beads and pipecleaners are optional.
- [ ] Print one **Binary Bracelets - Worksheet** per student.
- [ ] Print one **Binary Bracelets - Assessment** per student.
- [ ] Make sure every student has a **Think Spot Journal**.
- [ ] (Optional) Write a short message on the board in binary.
- [ ] Prepare to show the **Bits Versus Bytes - Student Video**.

Links

For the Teacher

- **Binary Bracelets** - Teacher Video
- **Binary Bracelets** - Lesson in Action Video
- **Binary Bracelets** - Worksheet
- **Binary Bracelets** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

For the Students

- **Bits Versus Bytes** - Student Video

Vocabulary

- **Binary** - A way of representing information using only two options.

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important word:

Binary - Say it with me: Bye-nair-ee

A way of representing information using only two options

Off and On

- If you've written a short message on the board in binary, call the students' attention to it and ask if anyone knows what it is or what it means.
  - Put the message aside and move on to prepping for the activity.
- You can start by asking the class if they have ever seen inside a computer.
  - What's in there?
  - This is a good place to actually show them the inside of a computer (or pictures of the inside of a computer).

- Wires carry information through the machine in the form of electricity.

- The two options that a computer uses with respect to this electrical information are "off" and "on." Just like the lights in this room!
  - When computers represent information using only two options, it's called "Binary."
- That theme of two options doesn't stop when the information gets to its destination.

- Computers also *store* information using binary.
  - Binary isn't always off and on.
    - Hard Disk Drives store information using magnetic positive and magnetic negative.
    - DVDs store information as either reflective or non-reflective.
  - How do you suppose we can convert real-life things that we want to store in a computer into binary?
    - Let's start with letters.
    - Use the **Binary Bracelets - Worksheet** to show how a computer might represent capital letters.
      - This is a good time to mention that each spot where you have a binary option is called a "binary digit" or "bit" for short.
      - Ask if anyone knows what a grouping of eight bits is called (it's a byte.)
      - Fun fact: A grouping of four bits is called a nibble.
      - Watch the **Bits Versus Bytes - Student Video** (~1 minute)
    - Go over a few examples of converting letters into binary, then back.
    - Afterward, write an encoded letter and give the class a few seconds to figure out what it is.
    - When the class can figure out that encoded letter on their own, you can move on to the activity.

Main Activity (20 min)

Binary Bracelets - Worksheet

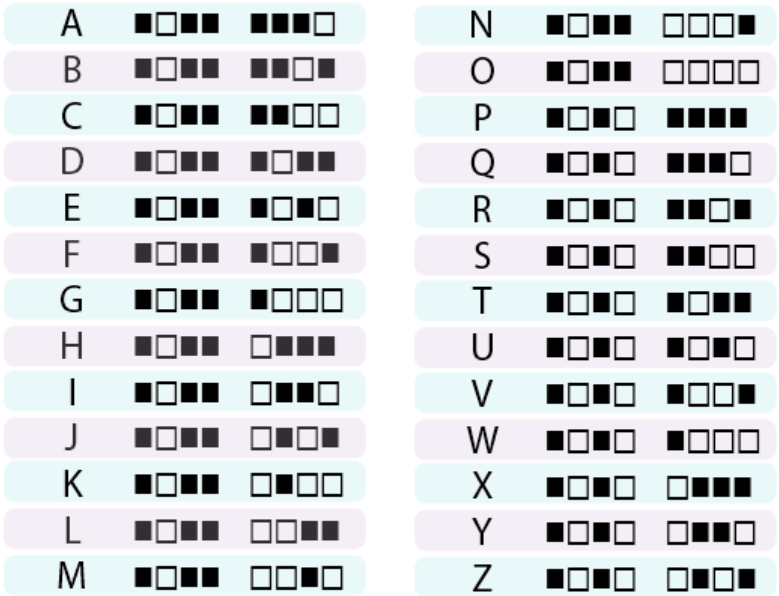
You do not need to cover the whole of binary, like counting and converting numbers back and forth from decimal. This lesson is intended to be a fun introduction to how computers store information, not a frustrating lesson in bases.

Lesson Tip

You know your classroom best. As the teacher, decide if students should do this individually or if students should work in pairs or small groups.

Directions:

- Find the first letter of your first name on the activity sheet.
- Fill in the squares of a bracelet to match the pattern of the squares next to the letter that you selected.
- Cut the bracelet out.
- Tape the bracelet around your wrist to wear it!
- Share your bracelet with your classmates to see if they can figure out your letter.



After the activity, revisit the message that was on the board and see if your class can decypher it using what they've learned.

Lesson Tip

If your class has extra budget for materials, try doing this exercise using thread (or pipe cleaners) and beads to create the binary bracelets instead of pen and paper. You can provide any combination of two colors in beads to the students, but black and white tend to be easiest, given the way that the key is done.

Wrap Up (5 min)

Flash Chat: What did we learn?

- What else do you think is represented as binary inside of a computer?
- How else might you represent binary instead of boxes that are filled or not filled?
- What was your favorite part about that activity?

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Use the activity worksheet to write out the rest of your name or your favorite word in binary.
- Imagine a world where we spoke in binary, saying "on" or "off", but nothing else. Draw two characters trying to talk to each other in binary.

Assessment (15 min)

Binary Bracelets - Assessment

- Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Binary Images

- There are several great resources on the web for taking this activity to the next level.
- If your students are interested in how images (or even music) can be represented as binary, you can find more details in Thinkersmith's **Binary Baubles**.

U

Unplugged

Name: _____ Date: _____

Binary Bracelets

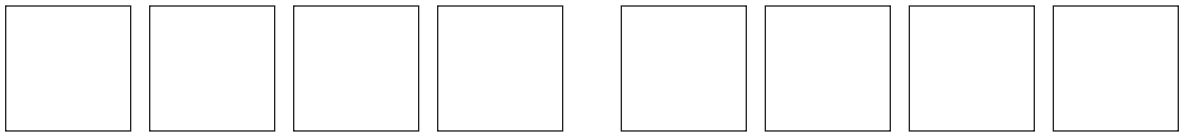
Binary Decoder Key

A	■□■■	■■■□
B	■□■■	■■□■
C	■□■■	■■□□
D	■□■■	■□■■
E	■□■■	■□■□
F	■□■■	■□□■
G	■□■■	■□□□
H	■□■■	□■■■
I	■□■■	□■■□
J	■□■■	□■□■
K	■□■■	□■□□
L	■□■■	□□■■
M	■□■■	□□■□
N	■□■■	□□□■
O	■□■■	□□□□
P	■□■□	■■■■
Q	■□■□	■■■□
R	■□■□	■■□■
S	■□■□	■■□□
T	■□■□	■□■■
U	■□■□	■□■□
V	■□■□	■□□■
W	■□■□	■□□□
X	■□■□	□■■■
Y	■□■□	□■■□
Z	■□■□	□■□■

Find the first letter of your first name.

Fill in the squares of the bracelet below to match the pattern of the squares next to the letter that you found.

Cut the bracelet out and tape it around your wrist to wear it!



U

Unplugged

Name: _____ Date: _____

Binary Bracelets

Assessment for Binary Bracelets Lesson

C O D E

Use the Binary Decoder Key below to decode the message at the bottom of the sheet.

A	■□■■	■■■□	N	■□■■	□□□■
B	■□■■	■■□■	O	■□■■	□□□□
C	■□■■	■■□□	P	■□■□	■■■■
D	■□■■	■□■■	Q	■□■□	■■■□
E	■□■■	■□■□	R	■□■□	■■□■
F	■□■■	■□□■	S	■□■□	■■□□
G	■□■■	■□□□	T	■□■□	■□■■
H	■□■■	□■■■	U	■□■□	■□■□
I	■□■■	□■■□	V	■□■□	■□□■
J	■□■■	□■□■	W	■□■□	■□□□
K	■□■■	□■□□	X	■□■□	□■■■
L	■□■■	□□■■	Y	■□■□	□■■□
M	■□■■	□□■□	Z	■□■□	□■□■

Can you figure out what the message says?

■□■■ ■■□□ _____

■□■■ □□□□ _____

■□■■ ■□■■ _____

■□■■ ■□■□ _____

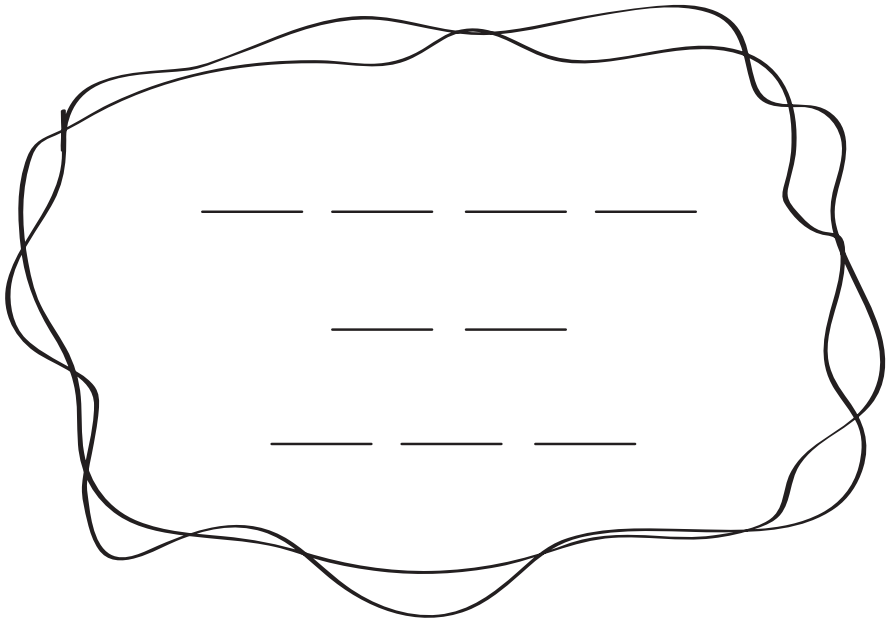
■□■■ □■■□ _____

■□■□ ■■□□ _____

■□■■ ■□□■ _____

■□■□ ■□■□ _____

■□■■ □□□■ _____



Course D

Lesson 1: Algorithms Unplugged: Graph Paper Programming

Unplugged | Programming | Program

Overview

By "programming" one another to draw pictures, students will begin to understand what coding is really about. The class will begin by having students instruct each other to color squares on graph paper in an effort to reproduce an existing picture. If there's time, the lesson can conclude with images that the students create themselves.

Purpose

The goal of this activity is to build critical thinking skills and excitement for the course.

By introducing basic concepts like *programming* and *algorithms* to the class in an unplugged activity, students who are intimidated by computers can still build a foundation of understanding on these topics. Programming and algorithms are essential to computer science. In this lesson, students will learn how to translate instructions into a program and recognize an algorithm.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Introduction to Graph Paper Programming**
- **Practice Together**

Main Activity (20 min)

- **Graph Paper Programming - Worksheet**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (10 min)

- **Graph Paper Programming - Assessment**

Extended Learning

Objectives

Students will be able to:

- Understand the difficulty of translating real problems into programs
- Explain how ideas may feel clear and yet still be misinterpreted by a computer
- Practice communicating ideas through codes and symbols

Preparation

- [ ] Watch the **Graph Paper Programming - Teacher Video**.
- [ ] Watch the **Graph Paper Programming - Lesson in Action Video**.
- [ ] Print out one **Graph Paper Programming - Worksheet** for each group.
- [ ] Print one **Graph Paper Programming - Assessment** for each student.
- [ ] Supply each group with several drawing grids, paper, and pens/pencils.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Graph Paper Programming** - Teacher Video
- **Graph Paper Programming** - Lesson in Action Video
- **Graph Paper Programming** - Worksheet
- **Graph Paper Programming** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has two new and important words:

- **Algorithm** - Say it with me: Al-go-ri-thm

A list of steps that you can follow to finish a task

- **Program** - Say it with me: Pro-gram

An algorithm that has been coded into something that can be run by a machine

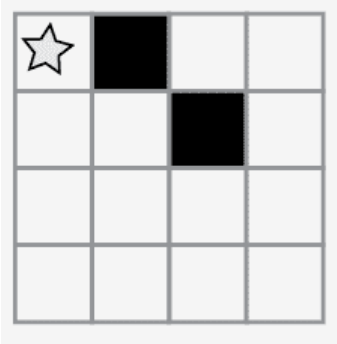
Introduction to Graph Paper Programming

In this activity, we are going to guide each other toward making drawings, without letting the other people in our group see the original image.

For this exercise, we will use sheets of 4x4 graph paper. Starting at the upper left-hand corner, we'll guide our teammates' Automatic Realization Machine (ARM) with simple instructions. Those instructions include:

- Move One Square Right
- Move One Square Left
- Move One Square Up
- Move One Square Down
- Fill-In Square with color

For example, here's how we would write an algorithm to instruct a friend (who is pretending to be a drawing machine) to color their blank grid so that it looks like the image below:

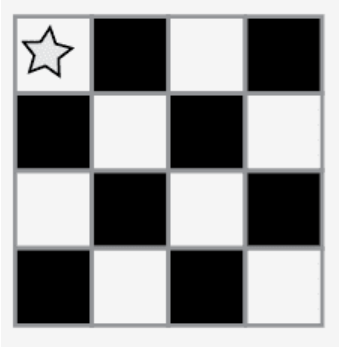


Lesson Tip

Have the class imagine that your arm is an Automatic Realization Machine (ARM). The idea of "algorithms" and "programs" will be brought to life even further if students feel like they're actually in control of your movements.

- Move One Square Right
- Fill In Square with Color
- Move One Square Right
- Move One Square Down
- Fill In Square with Color

That's simple enough, but it would take a lot of writing to provide instructions for a square like this:



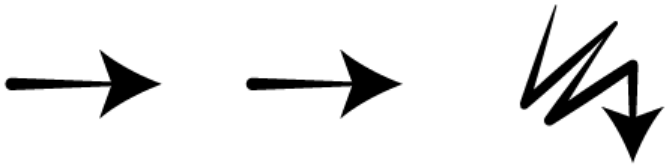
- Move One Square Right
- Fill In Square with Color
- Move One Square Right
- Move One Square Right
- Fill In Square with Color
- Move One Square Down
- Move One Square Left
- Fill In Square with Color
- Move One Square Left
- Move One Square Left
- Fill In Square with Color
- PLUS 12 MORE INSTRUCTIONS!

With one little substitution, we can do this much more easily! Instead of having to write out an entire phrase for each instruction, we can use arrows.

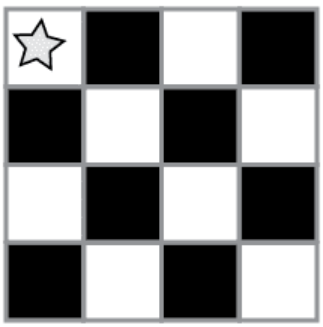
In this instance, the arrow symbols are the “program” code and the words are the “algorithm” piece. This means that we could write the algorithm:

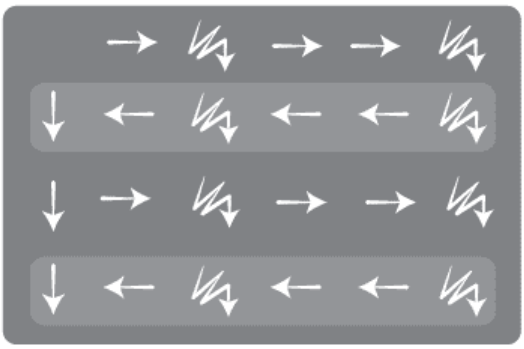
- “Move one square right, Move one square right, Fill-in square with color”

and that would correspond to the program:



Using arrows, we can redo the code from the previous image much more easily!
(Notice that we have written our program from left to right like you would read a book in English)



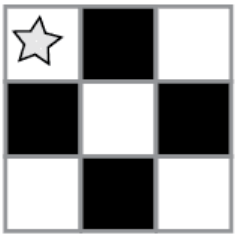


Ask the class to follow along with your finger and see if you can figure out how to get this image from the program to the right.

Practice Together

Start your class off in the world of programming by drawing or projecting the provided key onto the board.

Select a simple drawing, such as this one to use as an example.

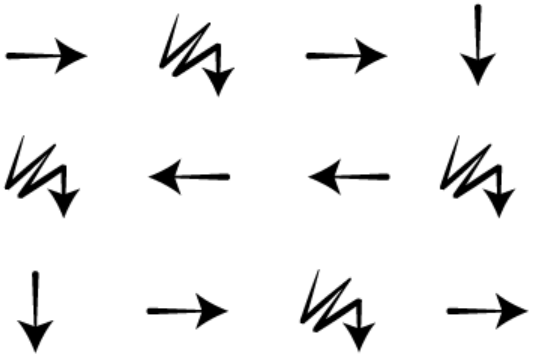


This is a good way to introduce all of the symbols in the key. To begin, fill in the graph for the class -- square by square -- then ask them to help describe what you've just done. First, you can speak the algorithm out loud, then you can turn your verbal instructions into a program.

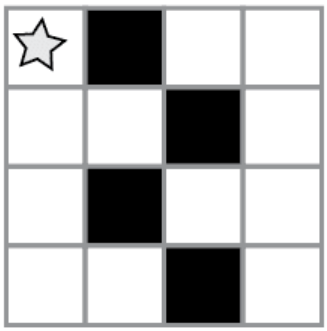
A sample algorithm:

"Move Right, Fill-In Square, Move Right, Move Down Fill-In Square, Move Left, Move Left, Fill-In Square Move Down, Move Right, Fill-In Square, Move Right"

Some of your class may notice that there is an unnecessary step, but hold them off until after the programming stage. Walk the class through translating the algorithm into the program:



The classroom may be buzzing with suggestions by this point. If the class gets the gist of the exercise, this is a good place to discuss alternate ways of filling out the same grid. If there is still confusion, save that piece for another day and work with another example.

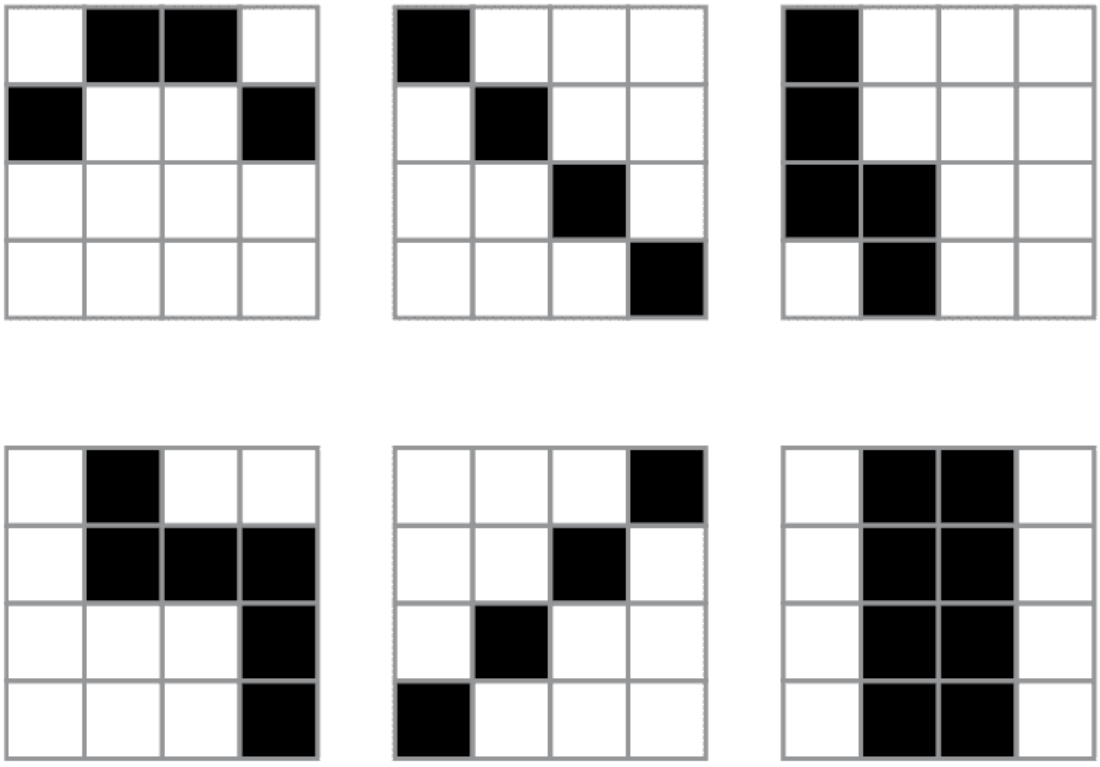


If the class can shout out the algorithm and define the correct symbols to use for each step, they're ready to move on. Depending on your class and their age, you can either try doing a more complicated grid together or skip straight to having them work in groups on their **Graph Paper Programming - Worksheet**.

Main Activity (20 min)

Graph Paper Programming - Worksheet

- Divide students into pairs.
  - Have each pair choose an image from the worksheet.
  - Discuss the algorithm to draw that image with partner.
  - Convert algorithm into a program using symbols.
  - Trade programs with another pair and draw one another's image.
  - Choose another image and go again!



Wrap Up (15 min)

Flash Chat: What did we learn?

- What did we learn today?
- What if we used the same arrows, but replaced "Fill-In Square" with "Lay Brick"? What might we be able to do?
- What else could we program if we just changed what the arrows meant?

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw another image that you could code. Can you write the program to make this drawing?

- What are some other instructions that might come in handy for big or complicated images?

Assessment (10 min)

Graph Paper Programming - Assessment

- Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

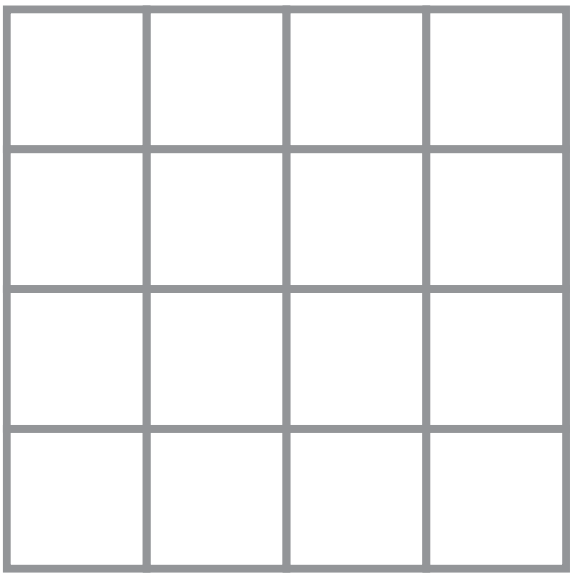
Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

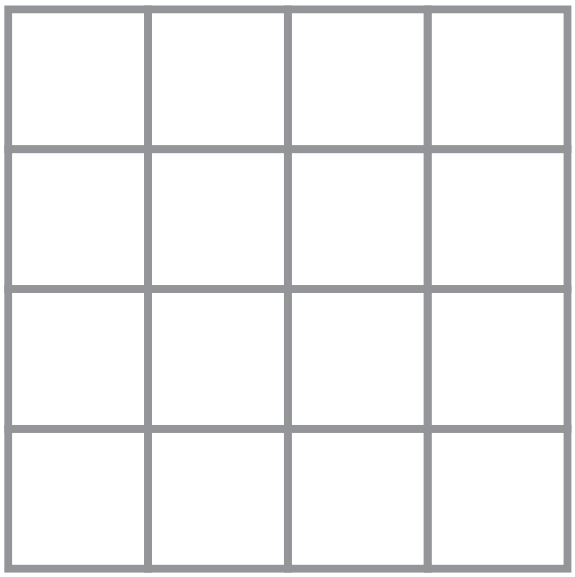
Better and Better

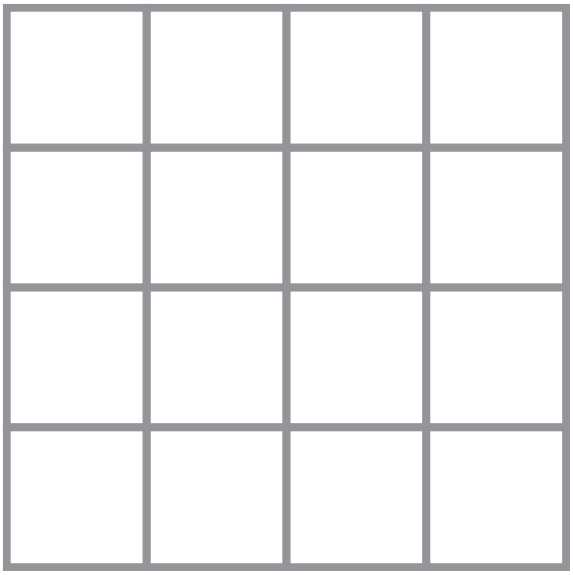
- Have your class try making up their own images.
- Can they figure out how to program the images that they create?

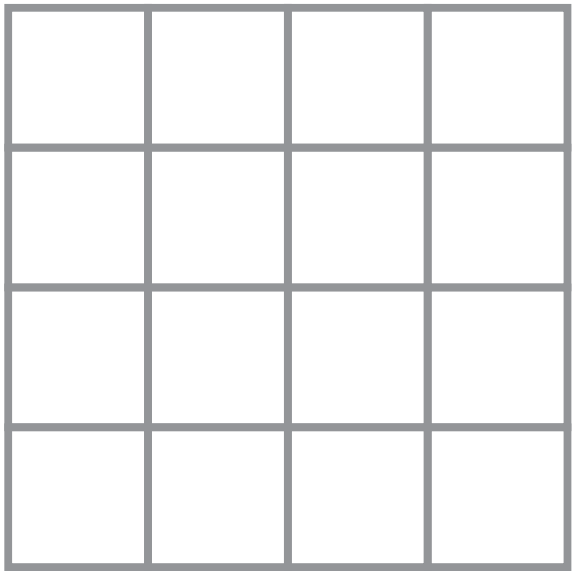
Class Challenge

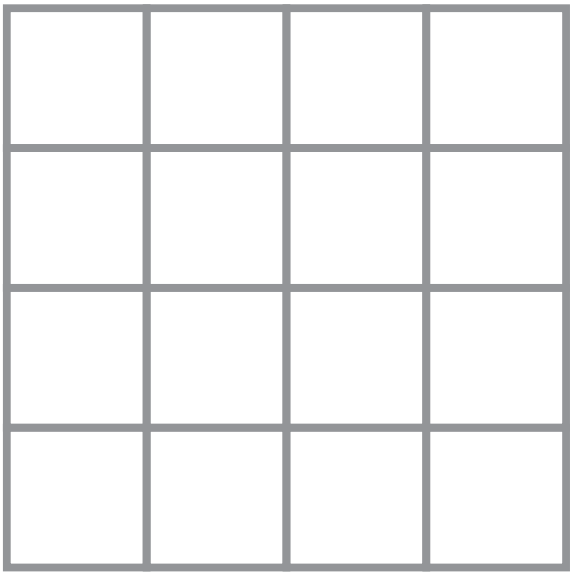
- As the teacher, draw an image on a 5x5 grid.
- Can the class code that up along with you?

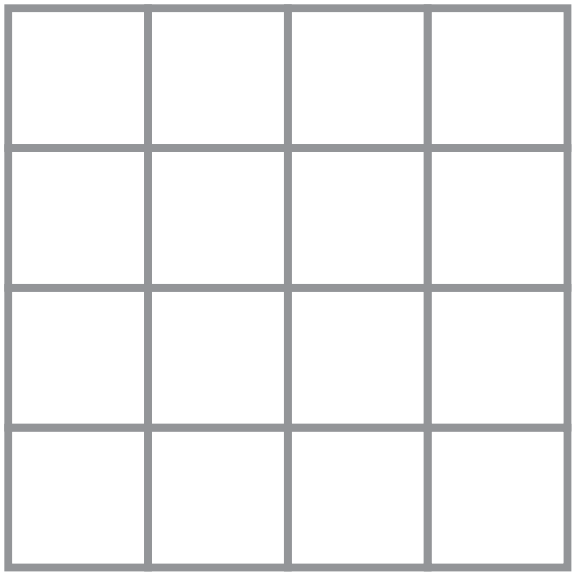


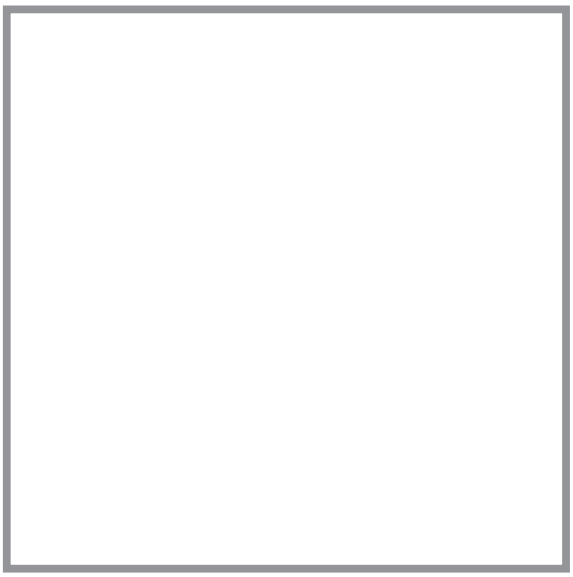
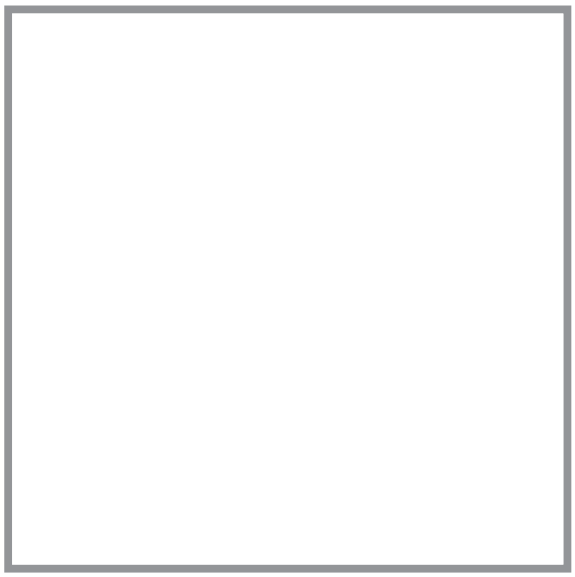
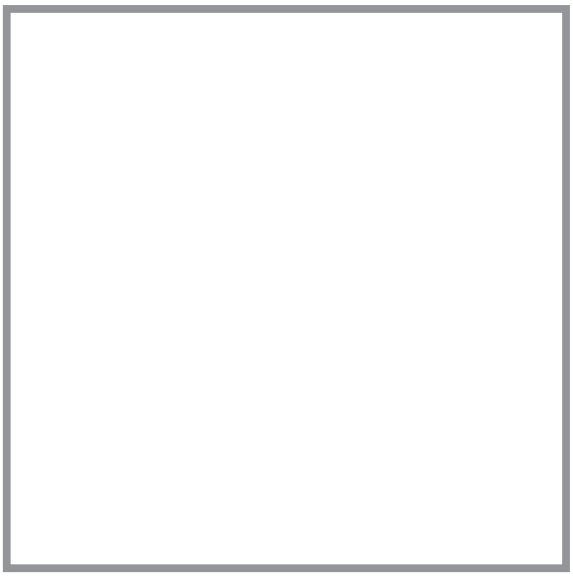
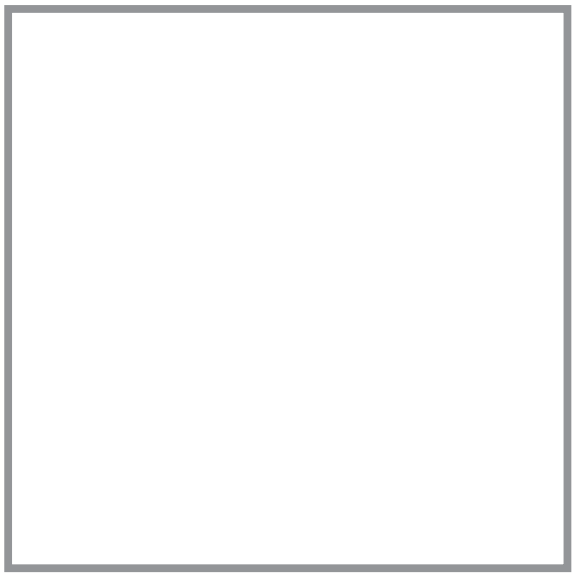
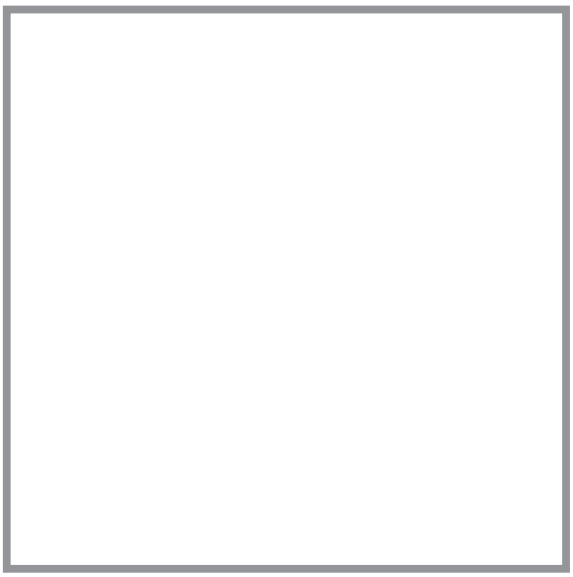
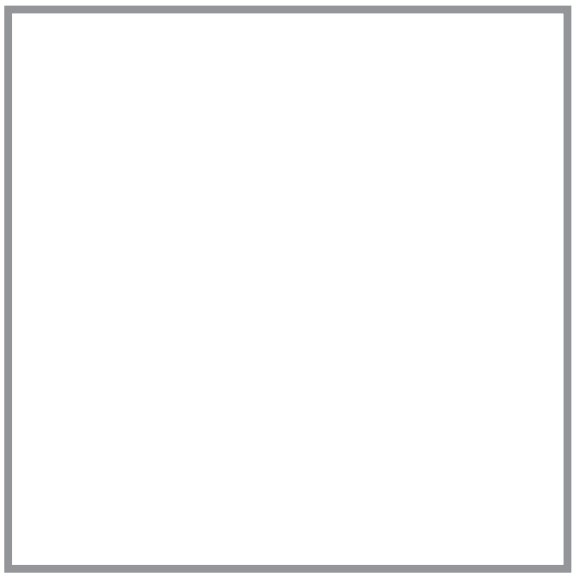










Name: _____ Date: _____

U

Unplugged

Graph Paper Programming

Four-by-Fours Activity Worksheet

C O
D E

Choose one of the drawings below to program for a friend. Don't let them see which one you choose!

Write the program on a piece of paper using arrows. Can they recreate your picture?

Use these symbols to write a program that would draw each image.


Move One Square Right


Move One Square Left


Move One Square Up


Move One Square Down


Fill-In Square with Color

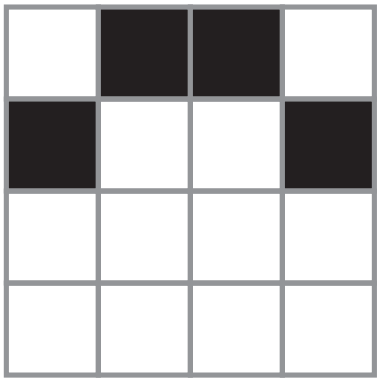


Image 1

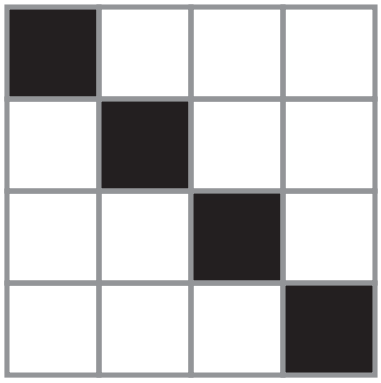


Image 2

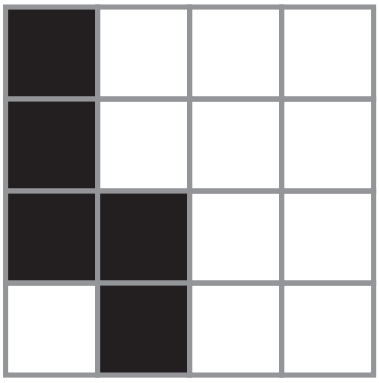


Image 3

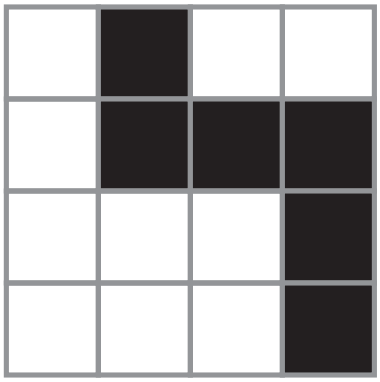


Image 4

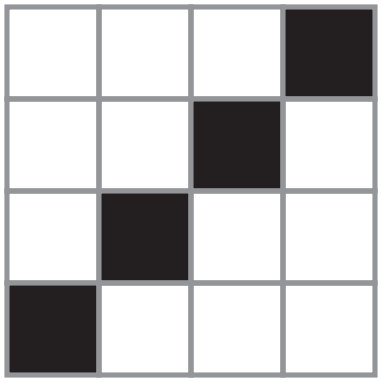


Image 5

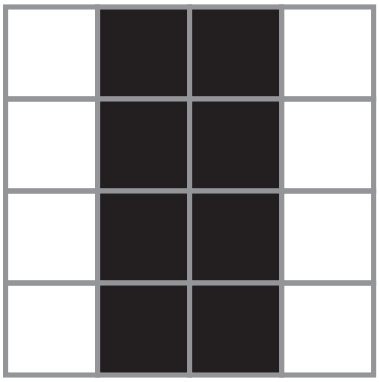


Image 6

Name: ____________________ Date: ____________

U
Unplugged

Graph Paper Programming

Assessment Worksheet

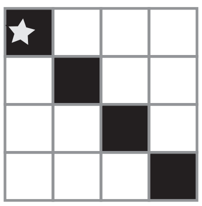
CODE

You have just learned how to create algorithms and programs from drawings, and how to draw an image from a program that someone gives to you. During the lesson, you worked with other people to complete your activities. Now you can use the drawings and programs below to practice by yourself.

Use the symbols below to write a program that would draw each image.

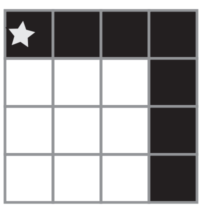
→	←	↑	↓	↯
Move One Square Forward	Move One Square Backward	Move One Square Up	Move One Square Down	Fill-In Square with Color

Start Here



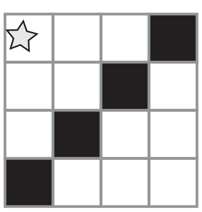
Step 1	2	3	4	5	6	7	8	9	10
Step 11	12	13	14	15	16	17	18	19	20

Start Here



Step 1	2	3	4	5	6	7	8	9	10
Step 11	12	13	14	15	16	17	18	19	20

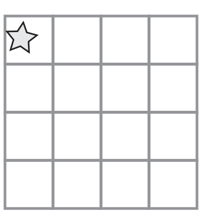
Start Here



Step 1	2	3	4	5	6	7	8	9	10
Step 11	12	13	14	15	16	17	18	19	20

Now, read the program below and draw the image that it describes.

Start Here



→ Step 1	↯ 2	→ 3	↓ 4	↯ 5	← 6	↓ 7	↯ 8	→ 9	↓ 10
↯ Step 11									

Lesson 7: Debugging Unplugged: Relay Programming

Unplugged | Relay Programming | Algorithms

Overview

This activity will begin with a short review of 'Graph Paper Programming', then will quickly move to a race against the clock, as students break into teams and work together to write a program one instruction at a time.

Purpose

There are many important components to this lesson. Students will be able to run around and get their wiggles out while building teamwork, programming, and debugging skills. Teamwork is very important in computer science. While **Pair Programming - Student Video** is common, it is more common for computer scientists to work in teams. These teams write and debug code as a group rather than individuals. In this lesson, students will learn to work together while being as efficient as possible.

This activity also provides a sense of urgency that will teach them to balance their time carefully and avoid mistakes, but not to fall too far behind.

Agenda

Warm Up (15 min)
- **Introduction**

Main Activity (15 min)
- **Relay Programming Activity**

Wrap Up (15 min)
- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (10 min)
- **Relay Programming - Assessment**

Extended Learning

Objectives

Students will be able to:

- Practice communicating ideas through code and symbols.
- Use teamwork to complete a task.
- Verify the work done by teammates to ensure a successful outcome.

Preparation

- [ ] Watch the **Relay Programming - Teacher Video**.
- [ ] Locate a wide open space for this activity, such as the gym or outdoor field.
- [ ] Print out one **Relay Programming Activity Packet - Teacher Prep Guide** for each group.
- [ ] Print one **Relay Programming - Assessment** for each student.
- [ ] Supply each group with plenty of paper and pens/pencils.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Relay Programming Activity Packet** - Teacher Prep Guide
- **Relay Programming** - Assessment
- **Relay Programming** - Teacher Video
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in your algorithm or program.

Teaching Guide

Warm Up (15 min)

Introduction

Recall that in 'Graph Paper Programming' we guided our teammate's Automatic Realization Machine (ARM) using arrows. Take a moment to go through a quick 'Graph Paper Programming' image as a reminder. It can either be one that you have already covered or one that is new.

We are going to do the same kind of thing today, but instead of controlling each other, we are going to work together to create a program one symbol at a time.

Main Activity (15 min)

Relay Programming Activity

Relay Programming Activity Packet - Teacher Prep Guide

The practice lesson was easy enough; let's add some action! We're going to do the same type of thing (create a program describing an image) but now we're going to do it in relay teams, one symbol at a time.

The rules of this game are simple:

- Divide students into groups of 3-5.
- Have each group queue up relay-style.
- Place an identical image at the other side of the room/gym/field from each team.
- Have the first student in line dash over to the image, review it, and write down the first symbol in the program to reproduce that image.
- The first student then runs back and tags the next person in line, then goes to the back of the queue.
- The next person in line dashes to the image, reviews the image, reviews the program that has already been written, then either debugs the program by crossing out an incorrect symbol, or adds a new one. That student then dashes back to tag the next person, and the process continues until one group has finished their program.

Clarifications

Here are some clarifications that need to be shared from time to time:

- Only one person from each group can be at the image at one time.
- It is okay to discuss algorithms with the rest of the group in line, even up to the point of planning who is going to write what when they get to the image.
- When a student debugs a program by crossing out an incorrect instruction (or a grouping of incorrect instructions) this counts as their entire turn. The next player will need to figure out how to correct the removed item.

First group to finish with a program that matches the image is the winner! Play through this several times, with images of increasing difficulty.

Wrap Up (15 min)

Flash Chat: What did we learn?

- What did we learn today?
- What if we were each able to do five arrows at a time?
  - How important would it be to debug our own work and the work of the programmer before us?
  - How about with 10 arrows?
  - 10,000? Would it be more or less important?
- Is it easier or harder to have multiple people working on the same program?
- Do you think people make more or fewer mistakes when they're in a hurry?
- If you find a mistake, do you have to throw out the entire program and start over?

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- How did teamwork play a role in the success of writing today's program?
- How did you use your debugging skills in today's lesson?

Assessment (10 min)

Relay Programming - Assessment

Pass around this assessment and have the students work on it independently. At the very end, you can take time to go over and discuss the answers.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Pass the paper

- If you don't have the time or room for a relay, you can have students pass the paper around their desk grouping, each writing one arrow before they move the paper along.

Fill It, Move It

- As the teacher, draw an image with as many filled squares as children in each group.
- Have the students write as many arrows in the program as it takes to get to a filled-in square (including actually filling that square in) before passing to the next person.

Debugging Together

Draw an image on the board. Have each student create a program for the image. Ask students to trade with their elbow partner and debug each other's code.

- Circle the first incorrect step, then pass it back.
- Give the students another chance to review and debug their own work.
- Ask for a volunteer to share their program.

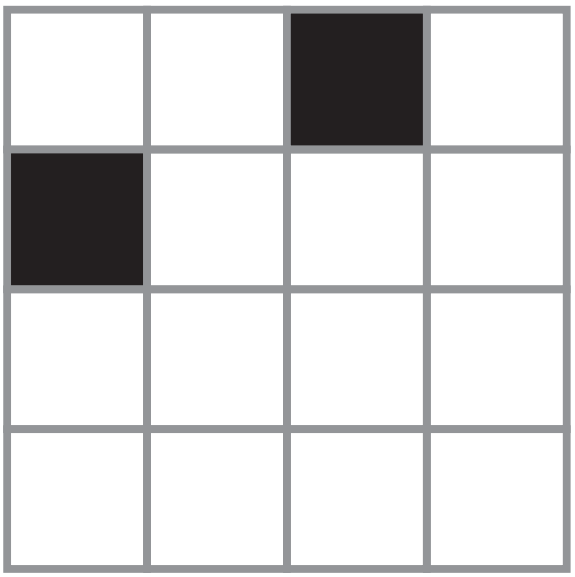
Ask the class:

- How many students had the same program?
- Anyone have something different?

1

Relay Programming

Relay Image 1

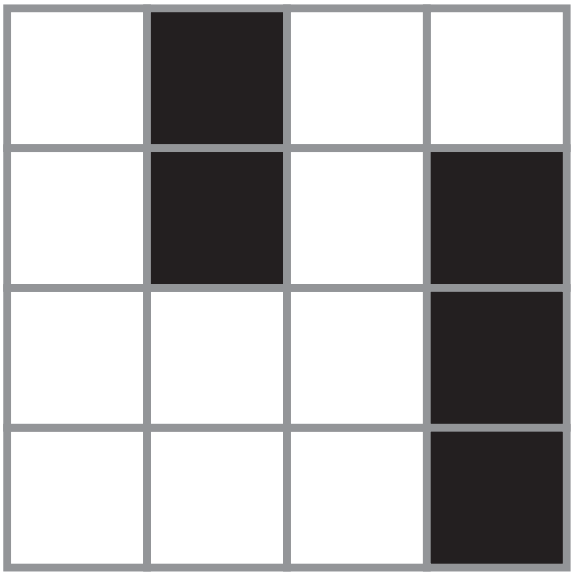


Revision 140710.1a

2

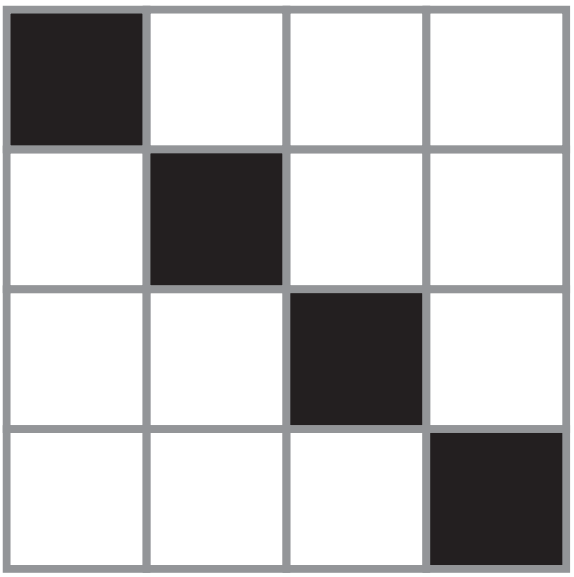
Relay Programming

Relay Image 2



3 Relay Programming

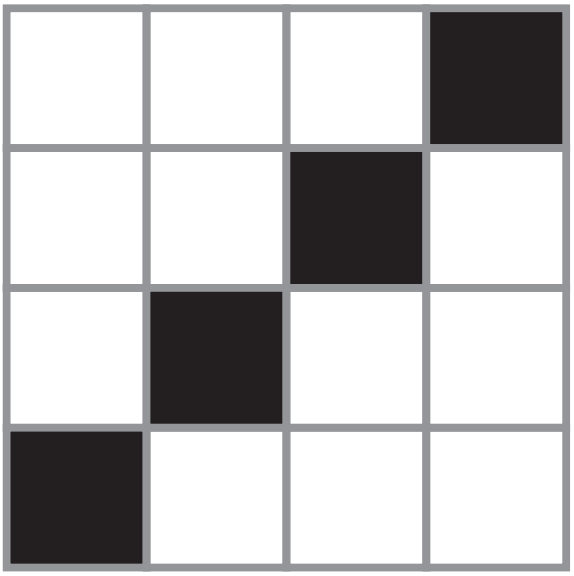
Relay Image 3



Revision 140710.1a

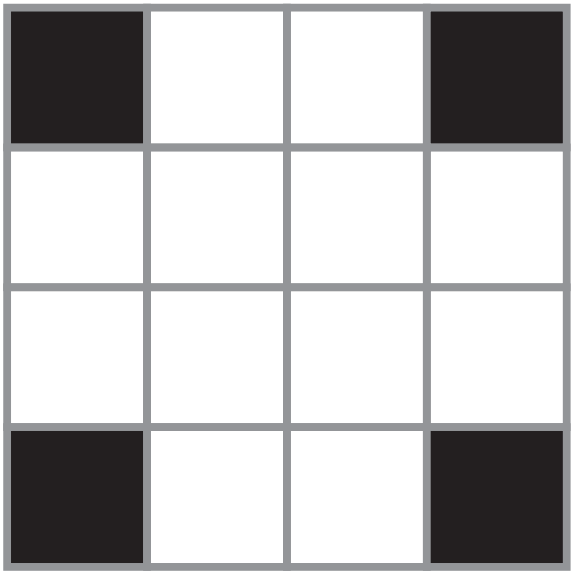
4 Relay Programming

Relay Image 4



5 Relay Programming

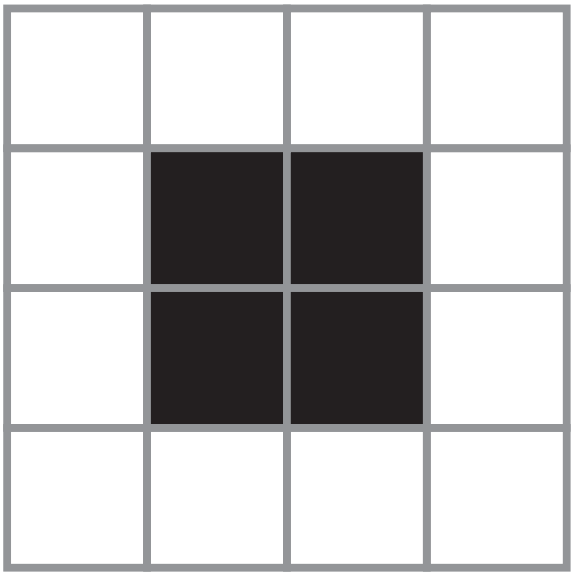
Relay Image 5

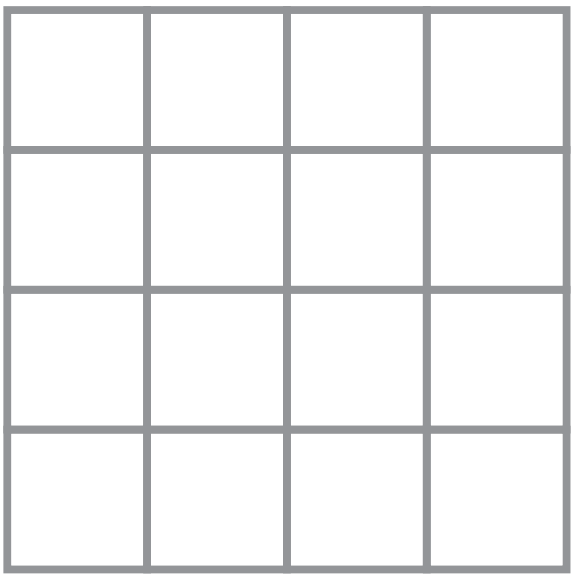


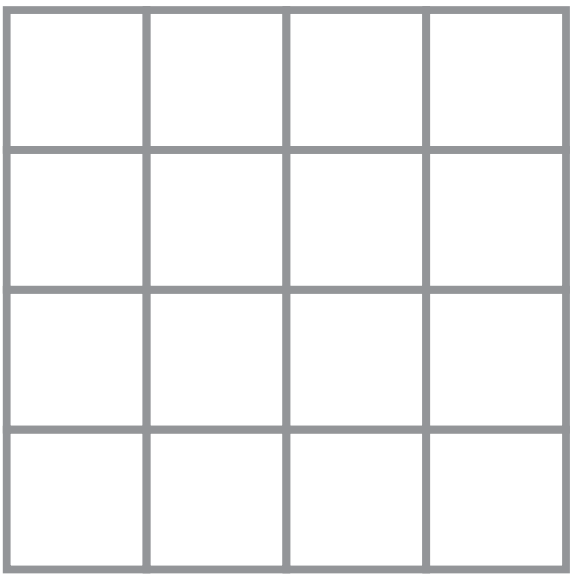
Revision 140710.1a

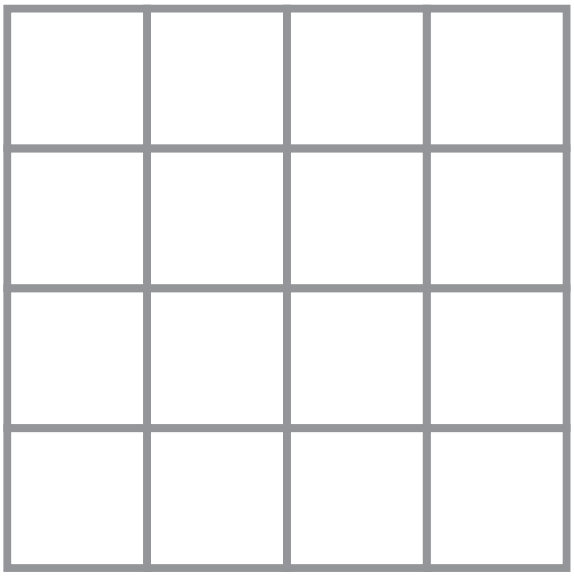
6 Relay Programming

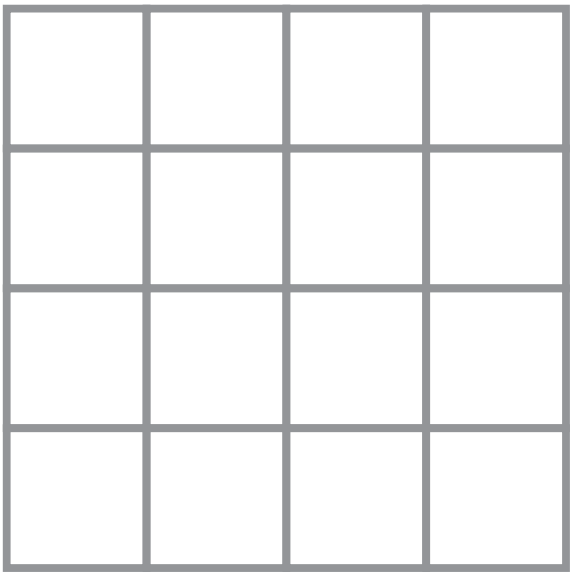
Relay Image 6

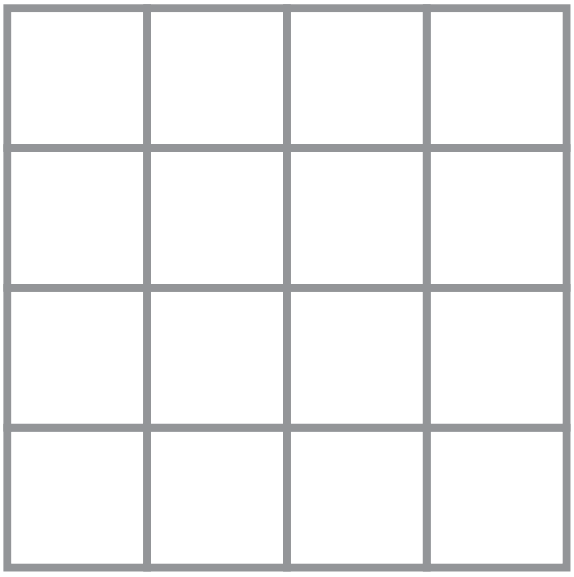


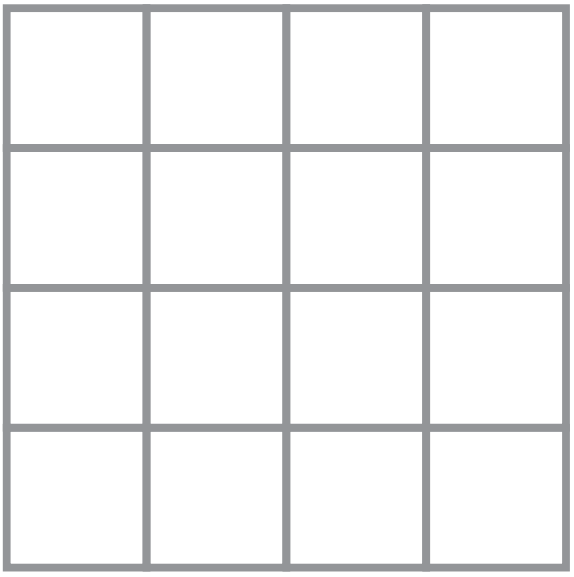


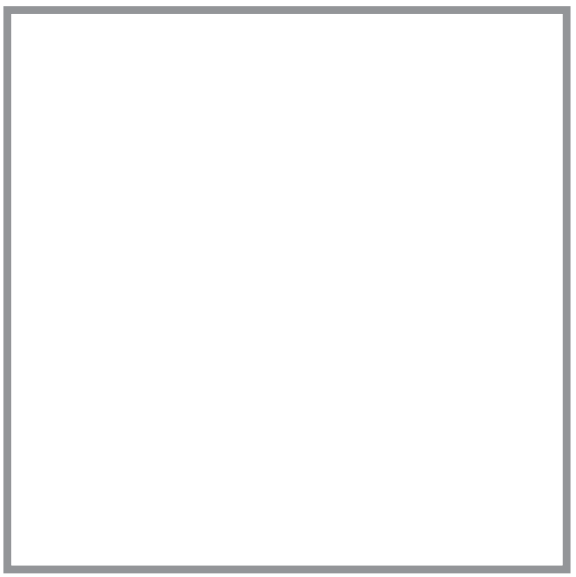


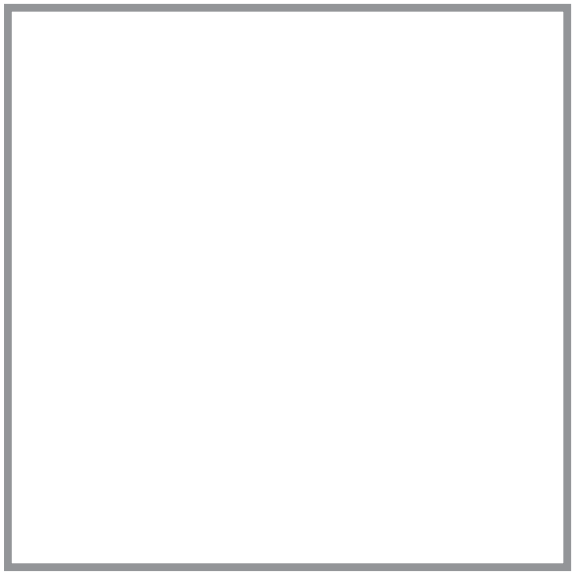


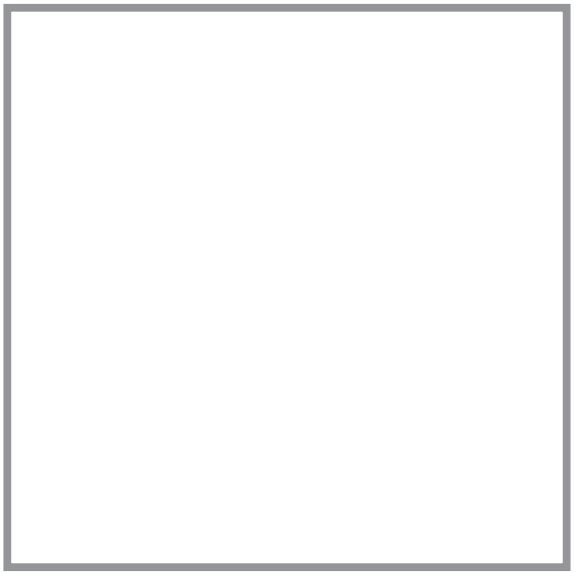


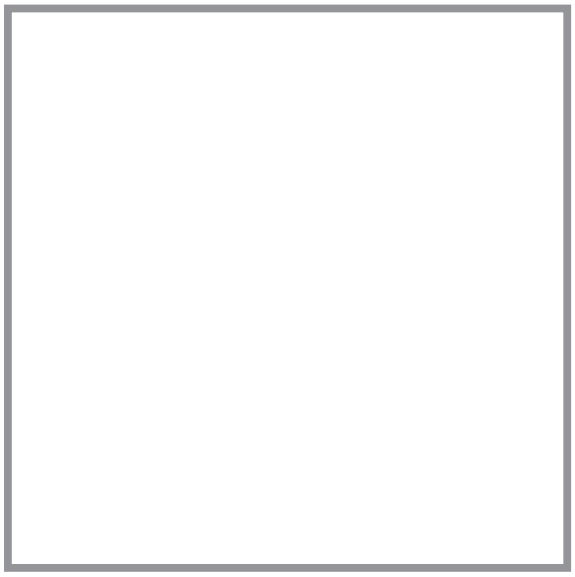


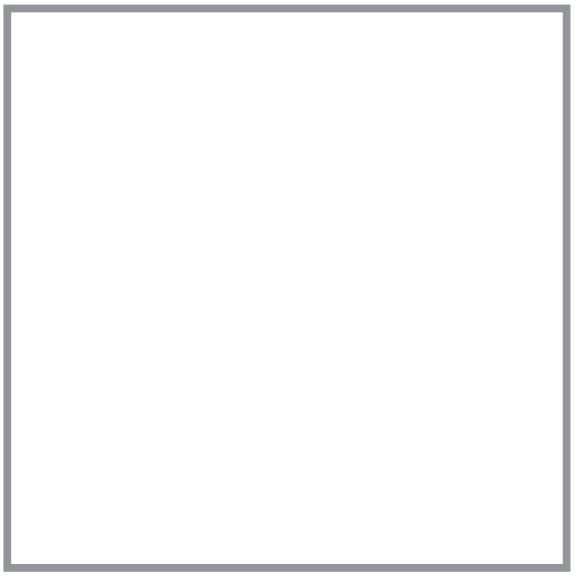












Name: _____ Date: _____

U

Unplugged

Debugging

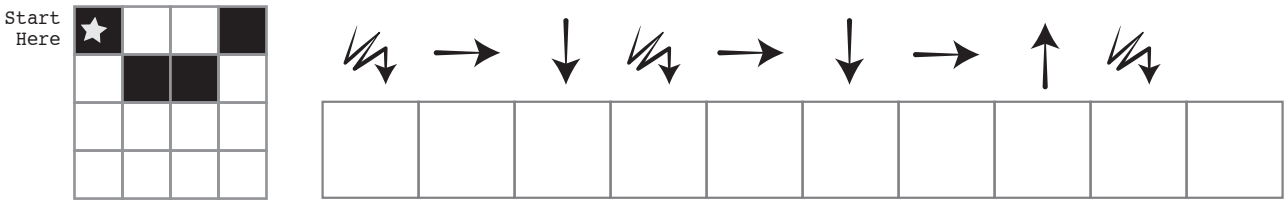
Assessment Worksheet

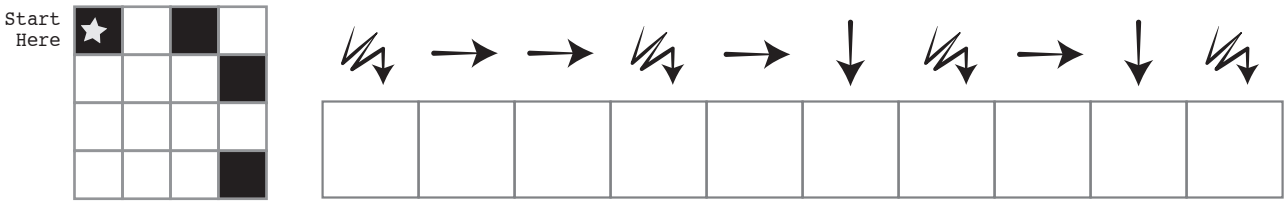
C O D E

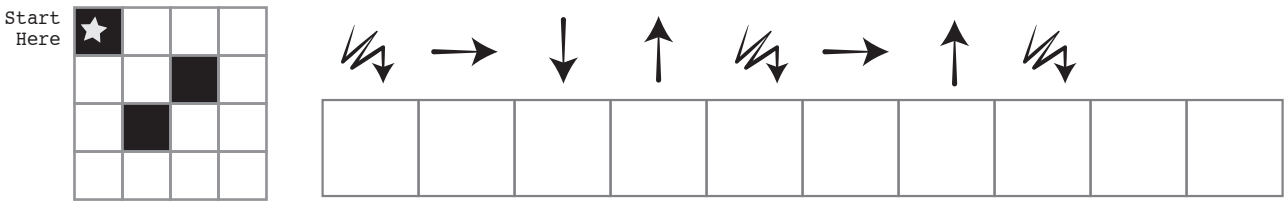
Sometimes when you are coding in groups, someone will make an error that will affect everyone.

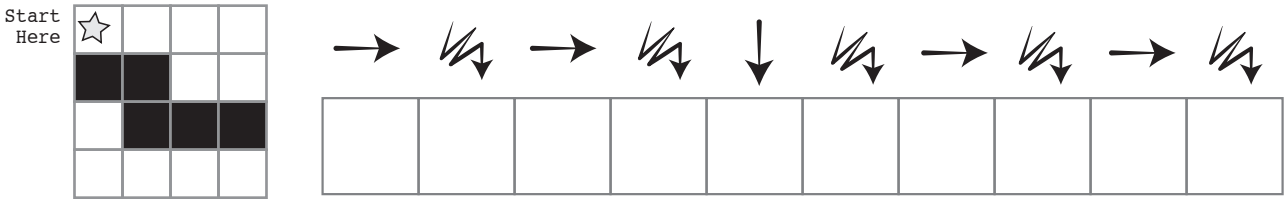
Somebody has already written programs for the images below, but each one has a mistake! Figure out what the programs are *supposed* to look like, and circle the error in each one. Then, draw the correct symbol in the box beneath.

Each program should use the symbols below to draw the image to its left.









Lesson 10: Conditionals with Cards

Conditionals | Unplugged

Overview

This lesson demonstrates how conditionals can be used to tailor a program to specific information. We don't always have all of the information we need when writing a program. Sometimes you will want to do something different in one situation than in another, even if you don't know what situation will be true when your code runs. That is where conditionals come in. Conditionals allow a computer to make a decision, based on the information that is true any time your code is run.

Purpose

One of the best parts of teaching *conditionals* is that students already understand the concept from their everyday lives.

This lesson merges computer science into the real world by building off of their ability to tell if a condition is true or false. Students will learn to use *if statements* to declare when a certain command should be run, as well as *if/else statements* to declare when a command should be run and what do run otherwise. Students may not recognize the word *conditionals*, but most students will understand the idea of using "if" to make sure that some action only occurs when it is supposed to.

Agenda

Warm Up (20 min)

- **Vocabulary Introduction**

Main Activity (20 min)

- **Conditionals with Cards Sample Program - Teacher Prep Guide**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (5 min)

- **Conditionals with Cards - Assessment**

Extended Learning

Objectives

Students will be able to:

- Define circumstances when certain parts of a program should run and when they shouldn't.
- Determine whether a conditional is met based on criteria.
- Traverse a program and predict the outcome, given a set of input.

Preparation

- [ ] Watch the **Conditionals With Cards - Teacher Video**.
- [ ] Watch the **Conditionals With Cards - Lesson in Action Video**.
- [ ] Gather decks of cards or something similar.
- [ ] One **Conditionals with Cards Sample Program - Teacher Prep Guide** for the class to look at.
- [ ] Print one **Conditionals with Cards - Assessment** for each student.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Conditionals With Cards** - Teacher Video
- **Conditionals With Cards** - Lesson in Action Video
- **Conditionals with Cards Sample Program** - Teacher Prep Guide
- **Conditionals with Cards** - Assessment
- **Conditionals With Cards** - Assessment Video
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Conditionals** - Statements that only run under certain conditions.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

Conditionals - Say it with me: Con-di-shun-uls

Statements that only run under certain conditions.

Introduction

- We can start this lesson off right away
  - Let the class know that if they can be completely quiet for thirty seconds, you will do something like:
    - Sing an opera song
    - Give five more minutes of recess
    - Do a handstand
  - Start counting right away.
  - If the students succeed, point out that they succeeded, so they get the reward.
  - Otherwise, point out that they were not completely quiet for a full thirty seconds, so they do not get the reward.
- Ask the class "What was the condition of the reward?"
  - The condition was IF you were quiet for 30 seconds
    - If you were, the condition would be true, and you would get the reward.
    - If you weren't, the condition would be false, so the reward woud not apply.
  - Can we come up with another conditional?
    - If you can guess my age correctly, the class can give you applause.
    - If I know an answer, I can raise my hand.
    - What examples can you come up with?
- Sometimes, we want to have an extra condition, in case the "IF" statement is not true.
  - This extra condition is called an "ELSE" statement
  - When the "IF" condition isn't met, we can look at the "ELSE" for what to do
    - Example: IF I draw a king from this deck of cards, everybody claps. Or ELSE, everyone says "Awwwwwwe."
    - Let's try it. (Draw a card and see if your class reacts appropriately.)
  - Ask the class to analyze what just happened.
    - What was the IF?
    - What was the ELSE?
    - Which condition was met?
  - Believe it or not, we have even one more option.
    - What if I wanted you to clap if I draw a 7, or else if I draw something less than seven you say "YAY," or else you say "Awwwwwwwe"?
      - This is why we have the terms If, Else-If, and Else.
      - If is the first condition
      - Else-If gets looked at only if the "If" isn't true.
      - Else gets looked at only if nothing before it is true.

Now let's play a game.

Main Activity (20 min)

Conditionals with Cards Sample Program - Teacher Prep Guide

Directions:

- Create a few programs with your class that depend on things like a card's suit, color, or value to award or subtract points. You can write the program as an algorithm, pseudocode, or actual code.

Here is a sample algorithm:

```
if (CARD is RED)
    Award YOUR team 1 point

Else
    Award OTHER team 1 point
```

Here is a sample of the same program in pseudocode:

```
If (card.color == RED){
    points.yours = points.yours + 1;
}

Else {
    points.other = points.other + 1;
}
```

- Decide how you want to split your class into teams.
- Each team should have a pile of cards (at least as many cards as team members) nearby.
- Put one of your “Programs” up on the board for all to see.
- Have the teams take turns drawing cards and following the program to see how many points they score in each round.
- Play several times with several different programs to help the students really understand conditionals.

Once the class has had some practice, you can encourage students to nest conditionals inside one another:

```
If (CARD is RED){
    Award YOUR team 1 point

Else
    If (CARD is higher than 9)
        Award OTHER team 1 point
    Else
        Award YOUR team the same number of points on the card
```

Here is the same program in pseudocode:

```
If (card.color == RED ){
    points.yours = points.yours + 1;
}
Else {
    if (card.value > 9){
        points.other = points.other + 1;
    }
    Else {
        points.yours = points.yours + card.value;
    }
}
```

Wrap Up (15 min)

Flash Chat: What did we learn?

- If you were going to code this up in Blocky, what would you need to add around your conditionals to let the code run more than one time? (A loop)
- What other things do you do during the day under certain conditions?
- If you are supposed to do something when the value of a card is more than 5, and you draw a 5, do you meet that condition?
- Notice that conditions are either "True" or "False." There is no assessment of a condition that evaluates to "Banana."
- When you need to meet several combinations of conditions, we can use something called "nested conditionals."
  - What do you think that means?
  - Can you give an example of where we saw that during the game?
- What part of that game did you like the best?

Lesson Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a conditional? How did you use a conditional today?
- What are some of the conditionals you used today? Can you come up with some more that you would use with a deck of cards?

Assessment (5 min)

Conditionals with Cards - Assessment

Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities. Here's a **Conditionals With Cards - Assessment Video** to watch as a guide.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

True/False Tag

- Line students up as if to play **Red Light / Green Light**.
- Select one person to stand in front as the Caller.
- The Caller chooses a condition and asks everyone who meets that condition to take a step forward.
  - If you have a red belt, step forward.
  - If you are wearing sandals, take a step forward.
- Try switching it up by saying things like "If you are *not* blonde, step forward."

Nesting

- Break students up into pairs or small groups.
- Have them write if statements for playing cards on strips of paper, such as:
  - the suit is clubs
  - the color is red
- Have students create similar strips for outcomes.
  - Add one point
  - Subtract one point
- Once that's done, have students choose three of each type of strip and three playing cards, paying attention to the order selected.
- Using three pieces of paper, have students write three different programs using only the sets of strips that they selected, in any order.
  - Encourage students to put some if statements inside other if statements.
- Now, students should run through all three programs using the cards that they drew, in the same order for each program.
  - Did any two programs return the same answer?
  - Did any return something different?

Conditionals with Cards

Sample Program 1

Sample program as algorithm

```
If (CARD is RED)
        Award YOUR team 1 point

Else
        Award OTHER team 1 point
```

This program has you choose a card. If the card is red, your team gets a point. Else, the other team gets a point.

Sample program from above as pseudocode (like code, but in no particular language)

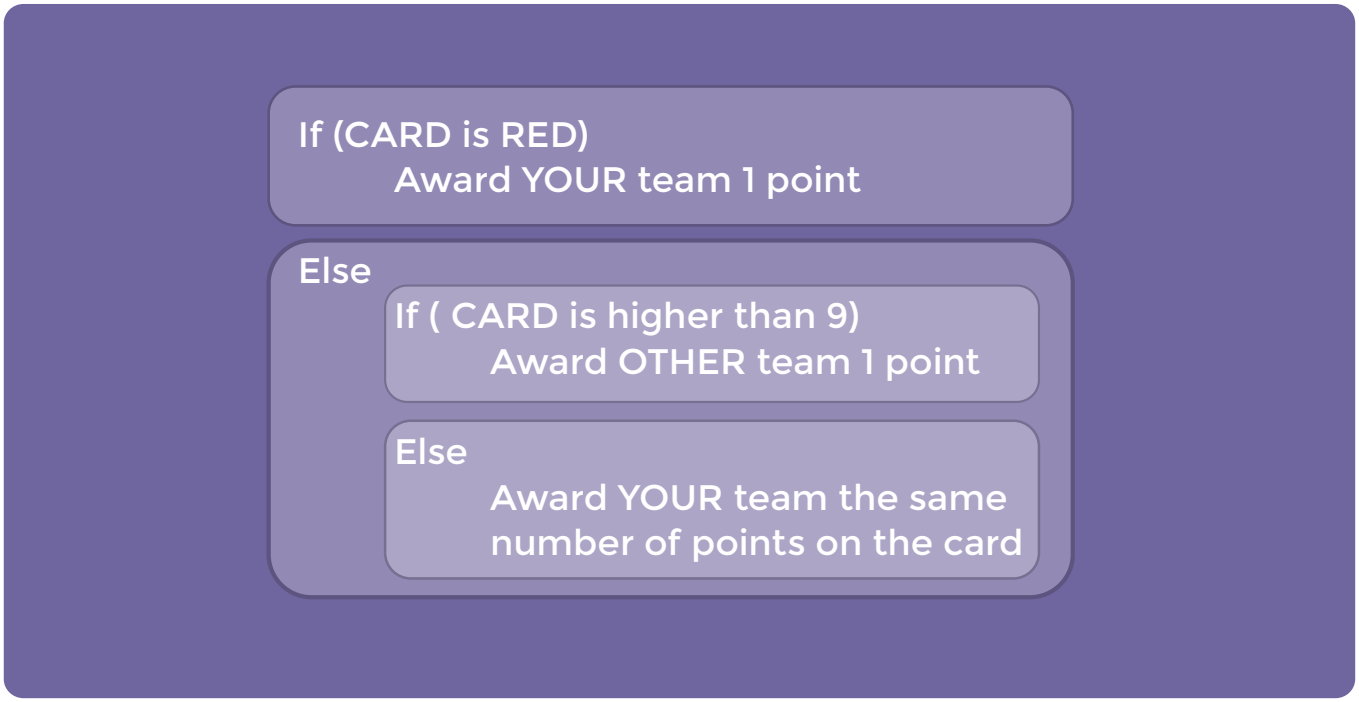
```
If (card.color == RED) {
        points.yours = points.yours + 1;
}

Else {
        points.other = points.other + 1;
}
```

Conditionals with Cards

Sample Program 2

Sample program as algorithm



This program has you choose a card. If the card is red, your team gets a point. Else, the card must be black. If your black card is higher than 9, then the other team gets a point, else your card must be black and lower than or equal to 9, and you get as many points as are on your card.

Sample program from above as pseudocode (like code, but in no particular language)

```
If (card.color == RED) {
        points.yours = points.yours + 1;
}

Else {
        If ( card.value > 9) {
                points.other = points.other + 1;
        }

        Else {
                points.yours = points.yours + card.value;
        }
}
```

Lesson 14: Common Sense Education: Digital Citizenship

Common Sense Education | Unplugged

Overview

In collaboration with **Common Sense Education - Website**, this lesson helps students learn to think critically about the user information that some websites request or require. Students learn the difference between private information and personal information, distinguishing what is safe and unsafe to share online.

Students will also explore what it means to be responsible and respectful to their offline and online communities as a step toward learning how to be good digital citizens.

Purpose

As students spend more time on computers, they should be aware that the internet is not always a safe space. In this lesson, students are taught what information is safe to share and what information should remain private. Students will create "superheros" and learn what it means to be a Digital Citizen on the internet.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **Personal vs. Private Online**

Main Activity (35 - 40 min)

- **Cubecraft Superhero Templates - Manipulatives**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (5 min)

- **Super Digital Citizen - Assessment**

Extended Learning

Objectives

Students will be able to:

- Compare and contrast their responsibilities to their online and offline communities.
- Understand what type of information can put them at risk for identity theft and other scams.
- Reflect on the characteristics that make someone an upstanding citizen.
- Devise resolutions to digital dilemmas.

Preparation

- [ ] Watch the **Digital Citizenship - Teacher Video**.
- [ ] Print out a good selection of male and female **Cubecraft Superhero Templates - Manipulatives** sheets for the whole class.
- [ ] Print one **Super Digital Citizen - Assessment** for each student.

Links

For the Teacher

- **Digital Citizenship** - Teacher Video
- **Cubecraft Superhero Templates** - Manipulatives
- **Super Digital Citizen** - Assessment
- **Common Sense Education** - Website
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Digital Citizen** - Someone who acts safely, responsibly, and respectfully online.

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important phrase:

Digital Citizen - Say it with me: Dih-jih-tal Sit-i-zen

Someone who acts safely, responsibly, and respectfully online

Personal vs. Private Online

- Ask "What types of information do you think are okay to share publicly online or on a profile that others will see?"
- What are some examples of websites where you must register in order to participate?
  - Write the names of the websites on the board.
- What information is required and why do you think it is required?
  - Information may be required to help distinguish one person from another.
  - The website may keep a record of who uses it.
- Explain that it's important to know that sharing some kinds of user information can put you and your family's privacy at risk.
- Point out that you do not have to fill out fields on websites if they are not required.
  - Required fields are usually marked by an asterisk (*) or are highlighted in red.
- Elementary school students should never register for sites that require private information without the approval and guidance of a parent or guardian.
- Here is an example of public versus private information:

Lesson Tip

If you have access to a computer, feel free to navigate to a site that might require this type of information, such as Gmail or Facebook.

SAFE - Personal Information	UNSAFE - Private Information
Your favorite food Your opinion (though it shoud be done respectfully) First name (with permission)	Mother's maiden name Social Security number Your date of birth Parents' credit card information Phone number

- Explain that some people will actively try to get you to share this kind of information so that they can use it to take over your identity. Once a thief has taken someone's identity, he or she can use that person's name to get a driver's license or buy things, even if the person whose identity they stole isn't old enough to do these things!
  - It's often not until much later that people realize that their identity has been stolen. Identity thieves may also apply for credit cards in other people's names and run up big bills that they don't pay off. Let students know that identity thieves often target children and teens because they have a clean credit history and their parents are unlikely to be aware that someone is taking on their child's identity.

Now, let's see what we can do to keep ourselves safe.

Main Activity (35 - 40 min)

Cubecraft Superhero Templates - Manipulatives

- Spiderman says "With great power comes great responsibility." This is also true when working or playing on the Internet.
- The things we read, see, and hear online can lead people to have all sorts of feelings (e.g., happy, hurt, excited, angry, curious).
  - What we do and say online can be powerful.
- The Internet allows us to learn about anything, talk to people at any time (no matter where they are in the world), and share our knowledge and creative projects with other people.
  - This also means that negative comments can spread very quickly to friends of all ages.
- CREATE a three-column chart with the terms “Safe,” “Responsible,” and “Respectful” written at the top of each column. Invite students to shout out words or phrases that describe how people can act safely, responsibly, and respectfully online, and then write them in the appropriate column.

Safe	Responsible	Respectful

Now, let's really make sure we understand how to be a Super Digital Citizen!

Directions:

- Have each student grab a small selection of papercraft sheets and encourage them to blend the pieces to make their very own super hero.
- Allow plenty of time for students to cut, glue, and color.
- Give students a 5 minute warning to wrap up.
- Separate students into groups of 2-4 and tell them to use their super heroes and leftover supplies to stage a scene in which one superhero sees an act of poor digital citizenship. Then have the superhero fix the problem … and save the day!
- Go around the room, having each student explain their scene to the class.

Lesson Tip

For more in-depth modules, you can find additions to this curriculum at the **Common Sense Education - Website** page on Scope and Sequence.

Wrap Up (15 min)

Flash Chat: What did we learn?

- What is a good way to act responsibly online?
- What kinds of personal information could you share about yourself without showing your identity?
- What kinds of superpowers or qualities did your digital superheroes have in common?
- What does Spider-Man’s motto “With great power comes great responsibility” mean to you, as someone who uses the internet?

Lesson Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Journaling

Having students write about what they learned, why it’s useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a Digital Citizen?
- What do you need to do to be a Digital Citizen?

Assessment (5 min)

Super Digital Citizen - Assessment

- Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Common Sense Education

- Visit **Common Sense Education - Website** to learn more about how you can keep your students safe in this digital age.

U

Unplugged

Name: _____ Date: _____

Digital Citizenship

Assessment Worksheet

C O D E

Just because you **can** do something online doesn't mean that you **should**!

Cross out the information that you should not share online. Use the words that are leftover as the key to what you should find in the word search.

WORDS

1) Your Credit Card Info (CARD)
2) Your Online Name (NICKNAME)
3) What You Ate Today (FOOD)
4) Your Email (EMAIL)
5) Your Favorite Color (COLOR)
6) The Last Book you Read (BOOK)
7) The School You Attend (SCHOOL)
8) Your Favorite Band (BAND)
9) Your Phone Number (PHONE)
10) Your Address (ADDRESS)
11) Your Birthday (BIRTHDAY)

D	N	L	M	W	U	R	E	C	D
U	F	I	D	V	V	H	C	O	N
C	J	Y	C	I	U	A	M	L	A
G	A	S	R	K	N	K	O	O	B
T	X	R	P	D	N	X	R	R	P
D	N	J	Y	X	I	A	I	B	T
E	O	R	N	X	I	E	M	W	P
D	K	O	Q	K	D	N	D	E	T
J	Z	C	F	O	B	I	K	E	G
W	P	V	C	I	Y	V	E	J	A

Write a paragraph in the area below, telling about what you will do when you're on the Internet to make sure that you practice kind and respectful behavior.

Lesson 16: Binary Images

Binary | Unplugged

Overview

Though many people think of binary as strictly zeros and ones, students will be introduced to the idea that information can be represented in a variety of binary options. This lesson takes that concept one step further as it illustrates how a computer can store even more complex information (such as images and colors) in binary, as well.

Purpose

In this lesson students will learn how information is represented in a way such that a computer can interpret and store it. When learning *binary*, students will have the opportunity to write code and share it with peers to view as images. This can then be related back to how computers read a program, translate it to binary, use the information in some way, then reply back in a way humans can understand. For example, when we type a sentence into a document then press "save", a computer translates the sentence into binary, stores the information, then posts a message indicating the document has been stored.

Agenda

Warm Up (10 min)

- **Vocabulary**
- **Introduction to Binary**

Main Activity (20 min)

- **Binary Images - Worksheet**

Wrap Up (10 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (10 min)

- **Binary Image - Assessment**

Extended Learning

Objectives

Students will be able to:

- Identify methods for encoding images into binary.
- Relate images to a peer using binary encoding.
- Reproduce an image, based on binary code.

Preparation

- Watch the **Binary Images - Teacher Video**.
- Print one **Binary Images - Worksheet** per pair.
- Print one **Binary Image - Assessment** per student.
- Gather groupings of items that can show opposites for students to use when coming up with their own binary encodings (Optional).
- Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Binary Images** - Teacher Video
- **Binary Images** - Worksheet
- **Binary Image** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Binary** - A way of representing information using only two options.
- **Binary Alphabet** - The two options used in your binary code.

Teaching Guide

Warm Up (10 min)

Vocabulary

This lesson has two new terms:

- **Binary** - Say it with me: Bi-nare-ee

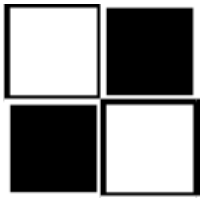
A way of representing information using only two options.

- **Binary Alphabet** - Say it with me: Bi-nare-ee Al-fa-bet

The two options used in your binary code.

Introduction to Binary

What if we had a picture like this, where there's only two color options for each square, black or white.



How might we encode this so that someone else could recreate the picture without seeing it?

- Some students might think back to the Graph Paper Programming lesson. While there could be a lot of similarities, let them know that this is different enough that they should not use that lesson to guide this one

You may hear suggestions like: "Say 'white, black, white, black'."

- "That's a great suggestion! Now I'm going to break you up into pairs. Work with your teammate to decide on a binary alphabet."

Decide whether you want your pairs to share their encodings with the other groups ahead of time, and tell them if they will be creating a key, or keeping their methods secret.

- "Now, let's encode some images, just like a computer would!"

Main Activity (20 min)

Binary Images - Worksheet

Now it's the students' turn!

Activity Directions:

1. Divide students into pairs.
2. Have them choose an image with their partner.
3. Encourage them to figure out what their binary alphabet is going to be.
4. Have them encode their image using their new binary alphabet.
5. Instruct students to trade encodings with another team and see if they can figure out which picture the other worked on.
6. Choose a Level
   - Easy: Let the other team know what your encoding method was
   - Tough: Have the other team guess your encoding method.

Wrap Up (10 min)

Flash Chat: What did we learn?

- What did we learn today?
- What kind of binary alphabet did you create?
- Can you think of how you could encode an image using only your fingers?
- Do you think you could create a binary alphabet out of sounds?

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a binary alphabet?
- What kind of information can you share using binary?

Assessment (10 min)

Binary Image - Assessment

Pass out this assessment for students to do individually. Try to save time at the end to go over answers.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Storing Color Images

- If your class really gets the idea behind storing binary images, they may want to know how to do color images.
  - First, you'll need to discuss how color works using binary (as in Binary Baubles, page 21).
  - Then, introduce some images that use combinations of those colors
- Encourage your students to come up with ways to code these color images.

Hexadecimal

- Take the idea of color one step further to introduce **hexadecimal color codes**.

Name: _____ Date: _____

Binary Images

Binary Representation Activity

Here are six images. Work with a partner to figure out how you can encode them into binary in such a way that another team can use the code to figure out what image you selected.

DIRECTIONS

1. Choose an image with your partner.
2. Figure out what your binary alphabet is going to be.
3. Encode your image using your new binary alphabet.
4. Trade your encoding with another team and see if you can figure out which picture they worked on.
5. Choose a Level
   * Easy: Let the other team know what your encoding method was
   * Tough: Have the other team guess your encoding method.

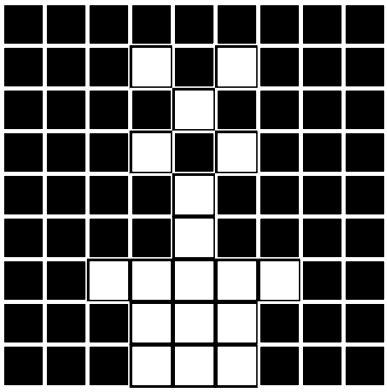


Image 1

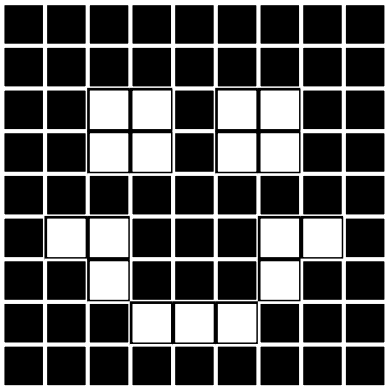


Image 2

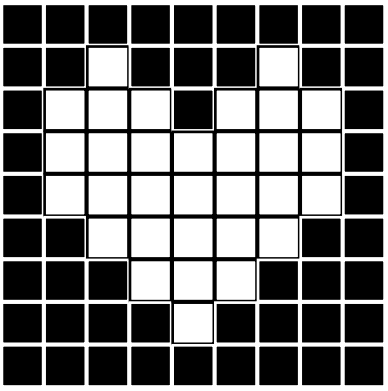


Image 3

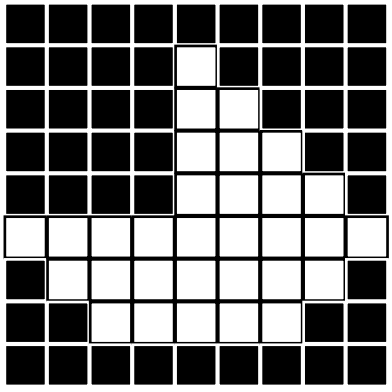


Image 4

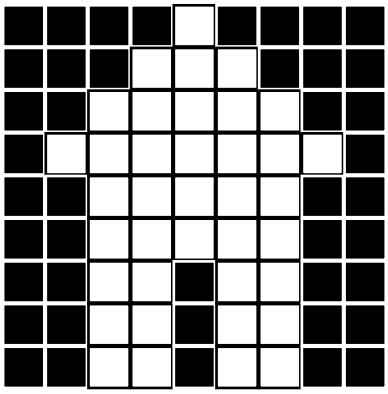


Image 5

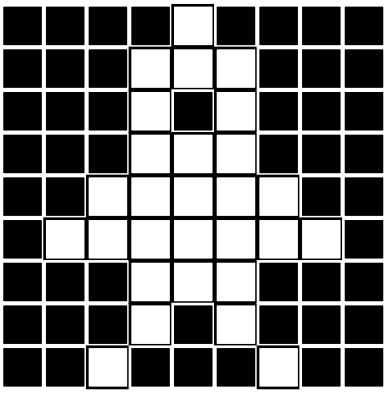


Image 6

Name: _____ Date: _____

Binary Images

Binary Representation Activity

Match the image to the binary code that describes it. In order to get the images correct, you will need to figure out the binary alphabet for each encoding.

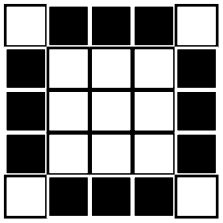


image #1

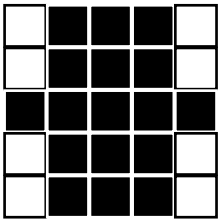


image #2

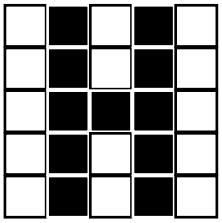


image #3

A) ★✖✖✖★★✖✖✖★✖✖✖✖✖★✖✖✖★★✖✖✖★

✖ = _____ ★ = _____ This encodes image # _____

B) ○●●●○●○○○●●○○○●●○○○●○●●●○

○ = _____ ● = _____ This encodes image # _____

C)

= _____ = _____ This encodes image # _____

How do you know that your answers are correct?

_____

_____

_____

_____

Course E

Lesson 1: Dice Race

Unplugged | Dice Race | Algorithms

Overview

In this lesson, students will relate the concept of algorithms back to real-life activities by playing the Dice Race game. The goal here is to start building the skills to translate real-world situations to online scenarios and vice versa.

Purpose

By introducing a basic concept like *algorithms* to the class in an unplugged activity, students who are intimidated by computers can still build a foundation of understanding on these topics. Algorithms are essential to computer science. In this lesson, students will learn how to translate instructions into a algorithm and how that plays a role in programming.

Agenda

Warm Up (15 min)

- **Introduction**
- **Vocabulary**

Main Activity (30 min)

- **Real-Life Algorithms: Dice Race - Worksheet**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Extended Learning

Objectives

Students will be able to:

- Decompose large activities into a series of smaller events.
- Arrange sequential events into their logical order.

Preparation

- [ ] Watch the **Real-Life Algorithms: Dice Race - Teacher Video**.
- [ ] Print one **Real-Life Algorithms: Dice Race - Worksheet** per group.
- [ ] Print one **Real-Life Algorithms: Dice Race - Assessment** per student.
- [ ] Give every student a **Think Spot Journal**.

Links

For the Teacher

- **Real-Life Algorithms: Dice Race** - Teacher Video
- **Real-Life Algorithms: Dice Race** - Worksheet
- **Real-Life Algorithms: Dice Race** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Algorithm** - A precise sequence of instructions for processes that can be executed by a computer

Teaching Guide

Warm Up (15 min)

Introduction

- Ask your students what they did to get ready for school this morning.
  - Write their answers on the board.
  - If possible, put numbers next to their responses to indicate the order that they happen.
    - If students give responses out of order, have them help you put them in some kind of logical order.
    - Point out places where order matters and places where it doesn't.
- Introduce students to the idea that it is possible to create algorithms for the things that we do everyday.
  - Give them a couple of examples, such as making breakfast, brushing teeth, planting a flower, and making paper airplanes.
- Computers need algorithms and programs to show them how to do even simple things that we can do without thinking about them.
  - It can be challenging to describe something that comes naturally in enough detail for a computer to replicate.
- Let's try doing this with a new and fun activity, like playing the Dice Race Game!

Vocabulary

This lesson has one vocabulary word that is important to review:

- **Algorithm** - Say it with me: Al-go-ri-thm

A list of steps to finish a task.

Main Activity (30 min)

Real-Life Algorithms: Dice Race - Worksheet

- You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to describe how we play the Dice Race Game.
- The hardest part about getting a problem ready for a computer can be figuring out how to describe real-life activities. We're going to get some practice by playing and describing the Dice Race game.

Lesson Tip

You know your classroom best. As the teacher, decide if students should do this in pairs or small groups.

Directions:

- Read the rules below.
- Play a couple rounds of the Dice Race game.
  - As you're playing, think about how you would describe everything that you're doing.
  - What would it look like from the computer's point of view?

Rules:

- Set each player's score to 0
- Have the first player roll

- Add points from that roll to player one's total score
- Have the next player roll
- Add points from that roll to player two's total score
- Each player should go again two more times
- Check each player's total score to see who has the most points
- Declare Winner

Lesson Tip

Help the students see the game from a computer's point of view. If they need to roll the dice, then the computer needs to provide dice. If the student needs to play three turns, then the computer needs to loop through the steps multiple times.

Game 1	Turn 1	Turn 2	Turn 3	Total
Player 1				
Player 2				

Circle the Winner!

Gather the class together and have each student complete the **Real-Life Algorithms: Dice Race - Assessment**. Once the students have completed the worksheet, have students share out their algorithms to the class. Open a discussion on the difference between an algorithm from a human's point of view and a computer's point of view.

Wrap Up (15 min)

Flash Chat: What did we learn?

- How many of you were able to follow your classmates' algorithms to play the Dice Race Game?
- What's the difference between an algorithm and a program?
  - An algorithm is the thinking behind what needs to happen, while the program is the actual instruction set that makes it happen.
  - An algorithm has to be translated into a program before a computer can run it.
- Did the exercise leave anything out?
  - What would you have added to make the algorithm even better?
  - What if the algorithm had been only one step: "Play Dice Race"?
    - Would it have been easier or harder?
    - What if it were forty steps?
- What was your favorite part about that activity?

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about
- How do you feel about today's lesson?
- What is an algorithm?
- What are some algorithms you use in your daily life?

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Go Figure

- Break the class up into teams.
- Have each team come up with several steps that they can think of to complete a task.
- Gather teams back together into one big group and have one team share their steps, without letting anyone know what the activity was that they had chosen.
- Allow the rest of the class to try to guess what activity the algorithm is for.

U

Unplugged

Name: _____ Date: _____

Real-Life Algorithms

Dice Race Activity

C O D E

You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to help each other understand the Dice Race game.

The hardest part about getting a problem ready for a computer can be figuring out how to describe real-life activities. We're going to get some practice by playing and describing the Dice Race game.

Read the rules below, then play a couple rounds of the Dice Race game. As you're playing, think about how you would describe everything that you're doing. What would it look like from the computer's point of view?

The Rules:

1) Set each player's score to 0.
2) Have the first player roll.
3) Add points from that roll to player one's total score.
4) Have the next player roll.
5) Add points from that roll to player two's total score.
6) Each player should go again two more times.
7) Check each player's total score to see who has the most points.
8) Declare Winner.

Game 1

	Turn 1	Turn 2	Turn 3	Total	
Player 1	_____	_____	_____	_____	Circle the Winner
Player 2	_____	_____	_____	_____	

Game 2

	Turn 1	Turn 2	Turn 3	Total	
Player 1	_____	_____	_____	_____	Circle the Winner
Player 2	_____	_____	_____	_____	

Dice Race

Assessment Worksheet

Use the space below to play through the Dice Race game.

When you're done, use the bottom of the page to create an algorithm (list of steps) that someone else could use to learn how to play.

	Turn 1	Turn 2	Turn 3	Total
Player 1				
Player 2				

Circle the Winner

Now, take the steps that you've used to play the game above, and write them down in the slots below. Take advantage of the repeat loop to avoid having to write down instructions more than once.

Step 1 : _____

Repeat 3 times

Step 2 : _____

Step 3 : _____

Step 4 : _____

Step 5 : _____

Step 6 : _____

Step 7 : _____

Lesson 4: Common Sense Education: Private and Personal Information

Common Sense Education | Personal Information | Private Information | Identity Theft

Overview

Developed by Common Sense Education, this lesson is about the difference between information that is safe to share online and information that is not.

As students visit sites that request information about their identities, they learn to adopt a critical inquiry process that empowers them to protect themselves and their families from identity theft. In this lesson, students learn to think critically about the user information that some websites request or require. They learn the difference between private information and personal information, as well as how to distinguish what is safe or unsafe to share online.

Purpose

Common Sense Education has created this lesson to teach kids the importance of security on the internet. By discussing the difference between personal and private information, students will be able to recognize what information should and shouldn't be shared. Students will also learn what signs you should look for to determine if a website is safe or not.

Agenda

Warm Up (5 min)

- **Introduction**

Main Activity (35 min)

- **Log In**
- **Private and Personal**
- **What's Safe to Share Online?**

Wrap Up (15 min)

- **Flash Chat: What did we learn about today?**
- **Journaling**

Assessment (10 min)

- **Private and Personal Information**

Objectives

Students will be able to:

- Learn about the benefits and risks of sharing information online.
- Understand what type of information can put them at risk for identity theft and other scams.

Preparation

- [ ] Copy the *Protect Yourself Student Handout* (7th page of the teacher prep guide), one for each student.
- [ ] Copy the *All About Me Student Handout* (6th page of the teacher prep guide), one for each student.
- [ ] Print out an assessment (8th page of the teacher prep guide) for each student. Teacher version is the page after the student assessment.
- [ ] Preview websites like **Neopets**, **Nickelodeon**, and **BookAdventure** and prepare to show them to the class.

Links

For the Teacher

- **Common Sense Education - Private and Personal Information** - Teacher Prep Guide
- **Common Sense Education** - Website
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Identity Theft** - When a thief steals someone's private information in order to pretend to be that person.
- **Personal Information** - Information that can't be used to identify you.
- **Private Information** - Information that can be used to identify you.
- **Register (Online)** - To enter your information in order to sign up and get access to a website.

Teaching Guide

Warm Up (5 min)

Introduction

Ask:

- What types of information do you think are okay to share publicly online such as on an online profile that others will see?
  - Interests and favorite activities
  - Opinions about a movie
  - First name
- What are some examples of websites where you must register in order to participate?
  - Social networking sites
  - Video-sharing sites
  - Youth discussion sites
  - Ask-an-expert sites
  - Game sites

Write the names of the websites on the board. Explain that it's important to know that sharing some kinds of user information can put you and your family's privacy at risk.

Main Activity (35 min)

common sense education®

Log In

Project for the class, or have students go online to **Neopets**, **Nickelodeon**, or **BookAdventure**. *Do not ask the students to sign up for these sites!*

Teacher Tip

As an offline alternative, print out and copy the website pages that ask for registration and log-in information. Distribute these to the students.

Discuss with the students the kinds of information that each website requires or requests before the users can participate.

Ask:

- What information is required? Why do you think it is required?
  - First name, username, password, password hint, gender, the state you live in, parent's permission, etc. This information is required because it helps distinguish one person from another. Or perhaps the website is keeping a record of who uses it.
- What information is optional? Why do you think it is optional?
  - Parent's email, birthday, state, country, gender, etc. This information is likely optional because the website does not require it for payment or to distinguish people. Or perhaps the website wants to keep track of this kind of information.
- Why do you think websites ask for this kind of information?
  - They want to get people to pay in order to use the site, they want to send messages to people who are signing up, or they want to try to sell things to those people.

Point Out that you do not have to fill out fields on websites if they are not required. Required fields are usually marked by an asterisk (*) or are highlighted in red.

Private and Personal

Explain to the students that some kinds of information are generally safe to share on the internet and some are not. However, the information that's considered safe should not be shared one-on-one with people the students don't already know offline.

Define:

- **Personal Information**: Information that can't be used to identify you.
- **Private Information**: Information that is about you, but can't be used to identify you.

Emphasize that personal information is usually safe to share online. Private information is usually unsafe to share online, meaning students should get permission from a parent or guardian before sharing this kind of information.

Teacher Tip

If you'd like a more clear distinction between "personal" and "private" information in these definitions, you can use other phrases like "friendly information" or "sharable information" to better define the line that the students should recognize. We chose to keep "personal" and "private" to stay true to Common Sense Education's lesson plan.

Share the following examples of information that is safe or unsafe to share:

SAFE - Personal Information	UNSAFE - Private Information
- Your favorite food - Your opinion (though it shoud be done respectfully) - First name (with permission)	- Mother's maiden name - Social Security number - Your date of birth - Parents' credit card information - Phone number

Ask:

- Why would someone want to steal someone else's identity on the internet?
  - To steal money
  - To do something bad or mean
  - To hide their real identity

Define:

- **Identity Theft**: When a thief steals someone's private information in order to pretend to be that person.

Explain that an identity thief uses private information to pretend to be the person whose identity he or she has stolen. Once the thief has taken someone's identity, he or she can use that person's name to get a driver's license or buy things, even if the person whose identity they stole isn't old enough to do these things! It's often not until much later that people realize their identity has been stolen. Identity thieves may also apply for credit cards in other people's names and run up big bills that they don't pay off. Let students know that identity thieves often target children and teens because they have a clean credit history and their parents are unlikely to be aware that someone is taking on their child's identity.

Emphasize the difference between private information (which can be used to steal your identity) and personal information (which cannot be used to steal your identity). Invite students to answer the following questions (write their answers on the board):

Ask:

- What kinds of private information could an identity thief use to find out and steal your identity?
  - First and last name, postal address, email address, phone numbers, passwords, credit card numbers, Social Security number, mother's maiden name.
- What kinds of personal information could you share about yourself without showing your identity?
  - Your age, gender, how many siblings you have, your favorite music, your favorite food, what pets you have, the name of your pet, your opinion about something.

Explain to students that on the internet, people you interact with could be your friends next door or strangers who live on the other side of the world. Because it's hard to know the intentions of people who you've never met before, it is best to remain cautious when sharing your information. You wouldn't give strangers your private information in the real world, and you need to be just as careful when you're online.

Remind students how important it is each time they share information online to stop and think: "Am I giving out information that I should keep private?" Point out that it can sometimes be safe to give out some private information. For example, a website might ask for your birth date or email address. But students should always ask their parent or guardian before giving out private

information.

Distribute the *Protect Yourself Student Handout* and have students complete the activity. Review the answers as a class.

What's Safe to Share Online?

Distribute the *All About Me Handout*. Have students write down all the personal information they would like to share on a public profile in an online community. Emphasize that even though personal information is safe to share online, it is okay to choose not to share it. Remind students that everything on the list should be safe to share; none of it should be private information that can put their identity at risk.

Encourage students to share their lists with the class.

Ask:

- Is there anything on the lists that could be used by an identity thief? Why?
  - Guide students to explain their answers and encourage them to use the vocabulary terms.

Wrap Up (15 min)

Flash Chat: What did we learn about today?

You can use these questions to assess your students' understanding of the lesson objectives. You may want to ask students to reflect in writing on one of the questions, using a journal or an online blog/wiki.

Ask:

- What is identity theft?
  - Using someone else's private information to pretend to be that person.
- How do personal information and private information differ?
  - Private information, such as a Social Security number, is unsafe to share. It should be kept private so that identity thieves cannot use it. Personal information, such as your favorite food, cannot be used by identity thieves and is safe to share. Even though personal information is usually safe to share online, you might choose not to share this information, and that's fine.
- What would be a good rule for kids about giving out private information?
  - They should not share it online without the permission of a teacher, parent, or guardian.

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What did you learn in today's lesson?
- How do you feel about today's lesson?
- Give an example of personal information and private information.
- What's a website that you use often? How do you know it is a safe website to use?

Assessment (10 min)

Private and Personal Information

Hand out the assessment to students. Allow students time to complete the assessment. If there is time left over, go over the answers with the students.

Private and Personal Information

Essential Question

How can you protect yourself from online identity theft?

Lesson Overview

As students visit sites that request information about their identites, they learn to adopt a critical inquiry process that empowers them to protect themselves and their families from identity theft. In this lesson, students learn to think critically about the user information that some websites request or require. They learn the difference between private information and personal information, distinguishing what is safe and unsafe to share online.

Learning Objectives

Students will be able to ...

- learn about the benefits of sharing information online, but also about the safety and security risks of sharing certain types of information.
- understand what type of information can put them at risk for identity theft and other scams.
- distinguish between personal information, which is safe to share online, and private information, which is unsafe to share.

Materials and Preparation

- Copy the **Protect Yourself Student Handout,** one for each student.
- Copy the **All About Me Student Handout,** one for each student.
- Preview the websites Neopets (www.neopets.com), Nickelodeon (www.nick.com), and BookAdventure (www.bookadventure.org) and be prepared to show them to the class.
- Chalkboard or white board

Family Resources

- Send home the **Online Security Family Tip Sheet (Elementary School)**.

Estimated time: 45 minutes

Standards Alignment –

Common Core:

grade 3: RI.1, RI.4, RI.10, RF.4a, W.4, W.7, W.10, SL.1a, SL.1b, SL.1c, SL.1d, SL.3, SL.4, SL.6, L.3a, L.6

grade 4: RI.1, RI.4, RI.10, RF.4a, W.4, W.7, W.10, SL.1a, SL.1b, SL.1c, SL.1d, SL.4, SL.6, L.3a, L.6

grade 5: RI.1, RI.4, RI.10, RF.4a, W.4, W.7, W.10, SL.1a, SL.1b, SL.1c, SL.1d, SL.4, SL.6, L.3a, L.6

NETS•S: 1b, 5a, 5b

Key Vocabulary –

register (online): to enter your information in order to sign up and get access to a website

personal information: information that can't be used to identify you, such as your age, gender, how many siblings you have, your favorite food, etc.

private information: information that can be used to identify you, such as your Social Security number, street address, email, phone number, etc.

identity theft: when a thief steals someone's private information in order to pretend to be that person

introduction

Warm-up *(5 minutes)*

ASK:

What types of information do you think are okay to share publicly online, on a profile that others will see, for instance?

Sample responses:

- Interests and favorite activities
- Opinions about a movie
- First name

INVITE students to share the names of websites they visit that require or request user information before allowing people to participate in online activities.

ASK:

What are some examples of websites where you must register in order to participate?

Review the Key Vocabulary term **register**.

Sample responses:

- Social networking sites
- Video-sharing sites
- Youth discussion sites
- Ask-an-expert sites
- Game sites

WRITE the names of the websites on the board. Explain that it's important to know that sharing some kinds of user information can put you and your family's privacy at risk.

teach 1

Log In *(15 minutes)*

PROJECT for the class, or have students go online to, www.neopets.com, www.nick.com, www. bookadventure.org, or one of the websites that your students suggested.

Note: As an offline alternative, print out and copy two of the website pages that ask for registration and log-in information. Distribute copies of the pages to each student.

DISCUSS with students the kinds of information that each website requires or requests before users can participate.

ASK:

What information is required and why do you think it is required?

This may include first name, user name, password, password hint, birth date, gender, the state you live in, parent's permission, etc. Let them know that the information may be required because it helps distinguish one person from another. Or perhaps the website is keeping a record of who uses it.

ASK:

What information is optional, and why do you think it is optional?	This may include parent's email, birthday, state, country, gender, etc. Maybe this information is optional because the website does not require it for payment, to distinguish people from one another, or so the website can keep track of this kind of information.
Why do you think websites ask for this kind of information?	Answers may include: They want to get people to pay in order to use the site, they want to send messages to people who are signing up, or they want to try to sell things to those people.

POINT OUT that you do not have to fill out fields on websites if they are not required. Required fields are usually marked by an asterisk (*) or are highlighted in red.

teach 2

Private and Personal *(10 minutes)*

EXPLAIN to students that some kinds of information are generally safe to share on the Internet and some are not. However, the information that's considered safe should not be shared one-on-one with people they don't already know offline.

DEFINE the Key Vocabulary terms **personal information** and **private information.** Emphasize that personal information is usually safe to share online. Private information is usually unsafe to share online (students should get permission from a parent or guardian).

SHARE the following examples of information that is safe or unsafe to share:

SAFE – Personal Information	UNSAFE – Private Information
• Your favorite food • Your opinion (though it should be done respectfully) • First name	• Mother's maiden name • Social Security number • Your date of birth • Parents' credit card information • Phone number

ASK:

Why would someone want to steal someone else's identity on the Internet?	Sample responses: • To steal money • To do something bad or mean • To hide their real identity

DEFINE Key Vocabulary term **identity theft**.

EXPLAIN that an identity thief uses private information to pretend to be the person whose identity he or she has stolen. Once the thief has taken someone's identity, he or she can use that person's name to get a driver's license or buy things, even if the person whose identity they stole isn't old enough to do these things! It's often not until much later that people realize their identity has been stolen. Identity thieves may also apply for credit cards in other people's names and run up big bills that they don't pay off. Let students know that identity thieves often target children and teens because they have a clean credit history and their parents are unlikely to be aware that someone is taking on their child's identity.

EMPHASIZE the difference between private information (which can be used to steal your identity) and personal information (which cannot be used to steal your identity). Invite students to answer the following questions (write their answers on the board):

ASK:

What kinds of private information could an identity thief use to find out and steal your identity?	Examples include: first and last name, postal address, email address, phone numbers, passwords, credit card numbers, Social Security number, mother's maiden name.
What kinds of personal information could you share about yourself without showing your identity?	Examples include: your age, gender, how many brothers and sisters you have, your favorite band, your favorite food, what pets you have, the name of your pet, your opinion about an important issue.

EXPLAIN to students that on the Internet, people you interact with could be your friends next door or strangers who live on the other side of the world. Because it's hard to know the intentions of people who you've never met before, it is best to remain cautious when sharing your information. You wouldn't give strangers your private information in the real world, and you need to be just as careful when you're online.

REMIND students how important it is each time they share information online to stop and think: "Am I giving out information that I should keep private?" Point out that it can sometimes be safe to give out some private information. For example, a website might ask for your birth date or email address. But students should always ask their parent or guardian before giving out private information.

DISTRIBUTE the **Protect Yourself Student Handout** and have students complete the activity. Review the correct answers (listed below):

- **Personal Information:** Your age, gender, how many brothers and sisters you have, your favorite band, your favorite food, the name of your pet.
- **Private Information:** Full name, street address, email address, your date of birth, phone numbers, credit card information, mother's maiden name, name of school.

teach 3

What's Safe to Share Online? *(10 minutes)*

DISTRIBUTE the **All About Me Student Handout**. Have students write down all the personal information they would like to share on a public profile in an online community. Emphasize that even though personal information is safe to share online, it is okay to choose not to share it. Remind students that everything on the list should be safe to share; none of it should be private information that can put their identity at risk.

ENCOURAGE students to share their lists with the class.

ASK:

Is there anything on the lists that could be used by an identity thief? Why?	Guide students to explain their answers and encourage them to use the Key Vocabulary terms.

closing

Wrap-up *(5 minutes)*

You can use these questions to assess your students' understanding of the lesson objectives. You may want to ask students to reflect in writing on one of the questions, using a journal or an online blog/wiki.

ASK:

What is identity theft?	Using someone else's private information to pretend to be that person.
How does personal information and private information differ?	Private information, such as a Social Security number, is unsafe to share. It should be kept private so that identity thieves cannot use it. Personal information, such as your favorite food, cannot be used by identity thieves and is safe to share. Even though personal information is usually safe to share online, you might choose not to share this information, and that's fine.
What would be a good rule for kids about giving out private information online?	They should not share it online without the permission of a teacher, parent, or guardian.

Private and Personal Information

Directions

Pretend you have a public profile in an online community. There are people you know, and people you don't know, who can view your profile.

Write down personal information that you would want to share about yourself. Make sure that everything on your list is safe to share and that it is not private information that could reveal your identity.

Use Common Sense!

Each time you share information about yourself online, stop and think: "Am I giving out information that I should keep private?"

Personal information often is safe to share. But you should never share private information without the permission of a parent, guardian, or teacher.

Private and Personal Information

Directions

Decide if each piece of information below is an example of personal information or private information. Then check the box to show your answer.

Information	Personal	Private
Full name (first and last)		
Age		
Street address		
Email address		
Date of birth		
Gender		
How many brothers and sisters you have		
Favorite band		
Phone numbers		
Credit card information		
Favorite food		
The name of your pet		
Mother's maiden name		
Name of your school		

Private and Personal Information

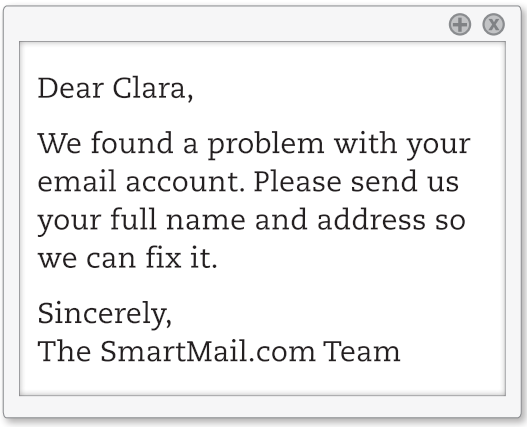
1. If a website asks you for your _____ online, you should talk to a parent or family member.

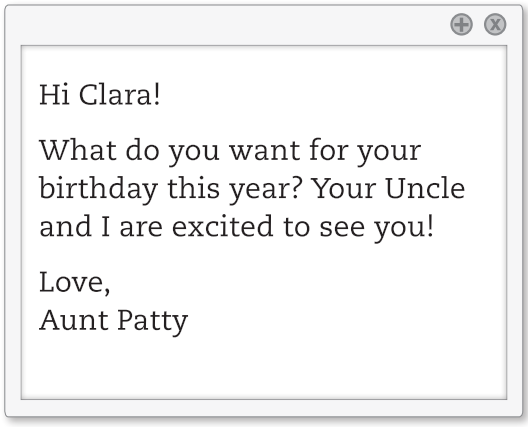
a) favorite color
b) date of birth
c) screen name

2. An identity thief probably would not be interested in your personal information, such as _____.

a) our full name
b) your street address
c) your favorite movie

3. Clara received two emails. Which email should she NOT respond to? Circle your answer.





Private and Personal Information

1. If a website asks you for your _____ online, you should talk to a parent or family member.

a) favorite color

b) date of birth

c) screen name

Answer feedback

The correct answer is **b**. Your date of birth is an example of private information. If a website asks for private information, you should ask a trusted adult before doing anything.

2. An identity thief probably would not be interested in your personal information, such as _____.

a) your full name

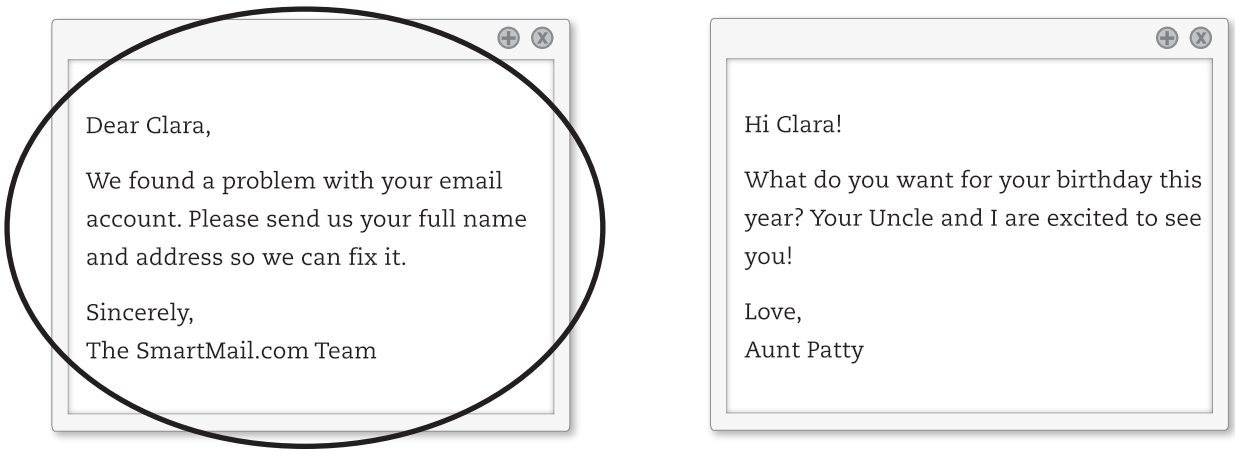
b) your street address

c) your favorite movie

Answer feedback

The correct answer is **c**. Both your full name and your street address are examples of private information. Personal information, like your favorite movie, would probably not be useful for an identity thief.

3. Clara received two emails. Which email should she NOT respond to? Circle your answer.



Answer feedback

If an email asks you for private information, such as your full name or address, you should not respond—especially if you do not know the person who sent the message.

Lesson 6: Functions Unplugged: Songwriting

Unplugged | Function

Overview

One of the most magnificent structures in the computer science world is the function. Functions (sometimes called procedures) are mini programs that you can use over and over inside of your bigger program. This lesson will help students intuitively understand why combining chunks of code into functions can be such a helpful practice.

Purpose

The use of functions helps simplify code and develop the student's ability to organize their program. Students will quickly recognize that writing functions can make their long programs easier to read and easier to debug if something goes wrong.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Sing a Song**

Main Activity (20 min)

- **Functions Unplugged: Songwriting - Worksheet**

Wrap Up (5 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (5 min)

- **Functions Unplugged: Songwriting - Assessment**

Extended Learning

Objectives

Students will be able to:

- Locate repeating phrases inside song lyrics.
- Identify sections of a song to pull into a function.
- Describe how functions can make programs easier to write.

Preparation

- [ ] Watch the **Functions Unplugged Songwriting - Teacher Video**.
- [ ] Watch the **Functions Unplugged: Songwriting - Lesson in Action Video**.
- [ ] Print several **Functions Unplugged: Songwriting - Worksheet** for each group.
- [ ] Print one **Functions Unplugged: Songwriting - Assessment** for each student.
- [ ] Access to the internet, or pre-downloaded songs and lyrics for activity.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Functions Unplugged Songwriting** - Teacher Video
- **Functions Unplugged: Songwriting** - Lesson in Action Video
- **Functions Unplugged: Songwriting** - Worksheet
- **Functions Unplugged: Songwriting** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Function** - A piece of code that you can easily call over and over again.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

- **Function** - Say it with me: Func-shun

A piece of code that you can call over and over again.

Sing a Song

- Let the class know that today is song day!
- We're going to learn a song together.
  - Start with a simple song, either written out or projected on the screen.
  - Point to the chorus and be sure that the class knows how it goes before you begin on the song.
  - Blast through the song, singing it with them in the beginning, then see what happens when you get to the part where it calls the chorus.

Chorus:

Little bunny Foo Foo
Hopping through the forest
Scooping up the field mice
And bopping ‘em on the head
Down came the Fairy
And she said
“Little bunny Foo Foo
I don’t wanna see you
Scooping up the field mice
And bopping ‘em on the head”*

Teaching Tip

Little Bunny Foo Foo is being used here as an example only. If your students know this song, feel free to use it. Otherwise, choose an appropriate song that they might be more familiar with (either from music class or the radio.)

Song:

Chorus

I’ll give you 3 chances.
Then I’ll turn you into a goon!
The next day. . .

Chorus

I’ll give you 2 chances.
Then I’ll turn you into a goon!
The next day. . .

Chorus

I’ll give you 1 chance.
Then I’ll turn you into a goon!
The next day. . .

Chorus

"I gave you two chances.
Now I’ll turn you into a goon!”
(POOF!)
And the moral of the story is:
Hare today, goon tomorrow!

- It's quite likely that the majority of the class will sing the lyrics for the chorus when you point to that bit.
  - Stop the song once that happens, and explicitly highlight what just happened.
    - You defined the chorus.
    - You called the chorus.
    - They sang the chorus.
- Ask the class why they suppose you only wrote the chorus once at the top of the paper instead of writing it over and over in each place where it is supposed to be sung.
  - What are other benefits of only writing the chorus once when you sing it many times?

Now, imagine that this song is a computer program. Defining a title (like "chorus") for a little piece of code that you use over and over again is called creating a function.

This is helpful to computer scientists for some of the same reasons that it is helpful to songwriters.

Lesson Tip

To hit this point home, you can look up the lyrics for some popular songs on the Internet. Show the students that the standard for repeating lyrics is to define the chorus at the top and call it from within the body of the song.

- It saves time not having to write all the code over and over in the program.
- If you make a mistake, you only have to change it one place.
- The program feels less complicated with the repeating pieces defined just once at the top.

We are going to play with songs a little more, to try to really understand how often this technique is used!

Main Activity (20 min)

Functions Unplugged: Songwriting - Worksheet

A fantastic way to compare functions to something we see in our everyday lives is to look at songs. Songs often have certain groups of lyrics that repeat over and over. We call such a group a "chorus."

Directions:

- Divide into groups of 4, 5, or 6.
- Give each group several copies of the Songwriting Worksheet.
- Play a short song for the class that contains a clear chorus that does not change from verse to verse.
- Challenge the class to identify (and write down) the chorus.
- Compare results from each group.

Did everyone get the same thing? Sing your choruses together to find out! Play this game over and over until the class has little trouble identifying the choruses.

Lesson Tip

It's most exciting for students to do this lesson with popular music from the radio, but if you're having a hard time finding appropriate songs where the lyrics repeat exactly, here are a few timeless options:

- **You Are My Sunshine**
- **Boom, Boom, Ain't it Great**
- **How Much Is That Doggie in the Window**
- **I Love Trash**

- It is often easier just to have the class listen to (or watch) the song, then vote on what the chorus is by singing it together, rather than writing the whole thing down. If you choose this method, consider having the class do a written chorus for the final song selection to be sure that the visual learners get proper reinforcement.

Wrap Up (5 min)

Flash Chat: What did we learn?

- Would you rather write lyrics over and over again or define a chorus?
- Do you think it's possible to make multiple choruses for the same song?
- Does it make sense to make a new chorus for every time it's needed in a song?

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a function and how do you use it?
- Can you think of another activity where you might want to call a special group of instructions several times?

> **Lesson Tip**
>
> Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Assessment (5 min)

Functions Unplugged: Songwriting - Assessment

Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Create Your Song

- Start by creating a chorus together, then repeat it between verses of a song that you develop around it.
- Make a change to the chorus, and ponder how much easier it is to change in just one place.
- Change the chorus again, making it much longer than it was originally.
- Add a second chorus and alternate between them in your verses.

Songwriting a Program

- What if we acted out songs instead of singing them? All of a sudden, our chorus would be a function of repeated actions, rather than words.
- Use the concepts of the arrows from the Graph Paper Programming lesson and create a program with lots of repeating instructions.
  - Circle those repeating actions so that the class can see where they are.
  - Define a function called "Chorus" above the program.
  - Cross out everywhere the repeating actions appear in the program and write "Chorus" instead.
- Repeat until the class can go through this process with little direction.

Name: _____ Date: _____

U

Unplugged

Songwriting

Using Lyrics to Explain Functions and Procedures

C O D E

One of the most magnificent structures in the computer science world is the function. Functions (sometimes called procedures) are mini programs that you can use over and over inside of your bigger program.

A fantastic way to compare functions to something we see in our everyday lives is to look at songs. Songs often have certain groups of lyrics that repeat over and over. We call such a group a "chorus."

Directions:

1) Divide into groups of 4, 5, or 6.

2) Give each group several copies of the Songwriting Worksheet.

3) Play a short song for the class that contains a clear chorus that does not change from verse to verse.

4) Challenge the class to identify (and write down) the chorus.

5) Compare results from each group. Did everyone get the same thing?

New Word!

Function

Say it with me: Func-shun

A piece of code that you can call over and over again.

Let's make a **function** for the bits that we use most often so that we don't need to write so much as we go.

Group Name: _____ Date: _____

U

Unplugged

Songwriting Worksheet Example

Using Lyrics to Explain Functions and Procedures

C O D E

Song Name:

Chorus:

Song Name:

Chorus:

Name: _____ Date: _____

U

Unplugged

Songwriting

Using Lyrics to Explain Functions - Assessment

C O D E

Look at the lyrics for the two songs below.

If it were your job to write these songs as computer programs, what chunk of code from each would you turn into a function so that you could use it over and over again with just one word?

Circle the segments of each program that repeat most often. Is everything that you circled exactly the same? If so, that can be your chorus!

Finish by writing the chorus for each song on the Songwriting Worksheet and give it a name. Those are your functions!

Song 1: I'm a Nut

I'm a little acorn brown
sitting on the cold, cold ground.
Everybody steps on me
that is why I'm cracked, you see.

I'm a nut
I'm a nut
I'm a nut, I'm a nut, I'm a nut

Called myself on the telephone
just to see if I was home.
Asked myself out on a date.
Picked me up at half-past eight.

I'm a nut
I'm a nut
I'm a nut, I'm a nut, I'm a nut

Took myself to the picture show.
Sat myself in the very first row.
Wrapped my arms around my waist.
Got so fresh, I slapped my face!

I'm a nut
I'm a nut
I'm a nut, I'm a nut, I'm a nut

Song 2: Skip to my Lou

Lou, Lou, skip to my Lou,
Lou, Lou, skip to my Lou,
Lou, Lou, skip to my Lou,
Skip to my Lou, my darlin'.

Fly's in the buttermilk,
Shoo, fly, shoo,
Fly's in the buttermilk,
Shoo, fly, shoo,
Fly's in the buttermilk,
Shoo, fly, shoo,
Skip to my Lou, my darlin'.

Lou, Lou, skip to my Lou,
Lou, Lou, skip to my Lou,
Lou, Lou, skip to my Lou,
Skip to my Lou, my darlin'.

Cows in the cornfield,
What'll I do?
Cows in the cornfield,
What'll I do?
Cows in the cornfield,
What'll I do?
Skip to my Lou, my darlin'.

Lou, Lou, skip to my Lou,
Lou, Lou, skip to my Lou,
Lou, Lou, skip to my Lou,
Skip to my Lou, my darlin'.

Group Name: _____ Date: _____

U

Unplugged

Songwriting Worksheet

Using Lyrics to Explain Functions - Assessment

C O D E

Song 1 Name:

Chorus:

Song 2 Name:

Chorus:

Lesson 16: The Internet

Overview

Even though many people use the internet daily, not very many know how it works. In this lesson, students will pretend to flow through the internet, all the while learning about connections, URLs, IP Addresses, and the DNS.

Purpose

If you have been doing every lesson in this course, then each student in your classroom has used the internet...but how many know how it works? Learning more about the internet will help students develop a better understanding of its endless possibilities.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Getting the Message**

Main Activity (20 min)

- **The Internet**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (5 min)

- **Internet - Assessment**

Objectives

Students will be able to:

- Learn about the complexity of sending messages over the internet.
- Translate URLs into IP Addresses.

Preparation

- [ ] Watch the **Internet - Teacher Video**.
- [ ] Print enough **IP Address Cards and Delivery Type Cards - Manipulatives** for each group.
- [ ] Print one **Internet - Assessment** for each student.
- [ ] Access to the internet (such as **get-site-ip.com**).
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Internet** - Teacher Video
- **IP Address Cards and Delivery Type Cards** - Manipulatives
- **Internet** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **DNS** - The service that translates URLs to IP addresses.
- **DSL/Cable** - A method of sending information using telephone or television cables.
- **Fiber Optic Cable** - A connection that uses light to transmit information
- **Internet** - A group of computers and servers that are connected to each other.
- **IP Address** - A number assigned to any item that is connected to the Internet.
- **Packets** - Small chunks of information that have been carefully formed from larger chunks of information.
- **Servers** - Computers that exist only to provide things to others.
- **URL** - An easy-to-remember address for calling a web page (like www.code.org).
- **Wi-Fi** - A wireless method of sending information using radio waves.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has several new and important words:

- **IP Address** - Say it with me: I-P Add-ress

A number assigned to any item that is connected to the Internet

- **DNS (Domain Name Service)** - Say it with me: D-N-S

The service that translates URLs to IP addresses

- **URL (Universal Resource Locator)** - Say it with me: U-R-L

An easy-to-remember address for calling a web page (like www.code.org)

- **Internet** - Say it with me: In-ter-net

A group of computers and servers that are connected to each other

- **Servers** - Say it with me: Ser-vers

Computers that exist only to provide things to others

- **Fiber Optic Cable** - Say it with me: Fye-ber Op-tic Cay-bl

A connection that uses light to transmit information

- **Wi-Fi** - Say it with me: Wye-Fye

A wireless method of sending information using radio waves

- **DSL/Cable** - Say it with me: D-S-L / Cay-bl

A method of sending information using telephone or television cables

- **Packets** - Say it with me: Pack-ets

Small chunks of information that have been carefully formed from larger chunks of information

> Lesson Tip
>
> A quick preview is all you need here. These words will all be explained as part of the lesson, so it would be far less confusing to do a brief intro to the words as a "see if you can spot these during the day" type of heads-up.

Getting the Message

- It's quite likely that your students are aware of what the internet is, but they may not really understand what the internet does.
  - Ask "What is the internet?"
  - Is the internet a public place or a private place?
  - (Truthfully, many people think it can be both, but it should be viewed as a public space no matter what settings you think you've mastered.)
  - How does information get from place to place?
- Let's say that I want to look at the webpage for Code.org. What do you suppose the process would be like for me to send a message to request that page?
  - What do I do as a user?
  - What do you think happens inside the internet?

> Lesson Tip
>
> There are some great YouTube videos on this subject that can make this lesson a little easier to understand. You can show them to the class in advance, or just watch them yourself. **Here is one of the most clear and entertaining versions**. (We recommend stopping the video at 2:59, if possible.)

Sending a message over the internet is a lot like sending a message through the mail...if every letter we sent required thousands of envelopes!

Every message we send through the internet gets chopped up and each piece is wrapped in its own version of an envelope. We call those "packets." Packets are specially formed chunks of information that are able to easily flow through any of the internet's channels.

Sometimes, a few of those packets will get lost, because the internet is a crazy place. In that case, the packets need to be resent, and the whole message has to get put on hold until they arrive.

Where do you think those packets are headed?

- Even if you're sending messages to another person, they first have to go to at least one "server."
  - A server is a special computer that is supposed to be always on and ready to send and receive information.
  - Every website has a server.
  - Even email goes through servers.

Servers don't have names like you and I do. They're actually addressed using numbers. These numbers are called IP addresses, and they look a little strange.

- For example: One of Code.org's IP addresses used to be 54.243.71.82
  - (Please be sure to check this out in advance. Most IP addresses change from time to time and they are then reused for other sites.)

There are many ways to reach the internet from your house, school, or place of business.

- You can connect directly using a cable (that might be DSL, Cable, or Fiber Optic)
- Or you can connect using radio waves over the air through Wi-Fi

Direct connections are most reliable, but they can be inconvenient.

- Can you figure out why?
  - (You have to be attached to a cable!)

Wi-Fi connections are super convenient, but the aren't always reliable.

- Can you figure out why not?
  - (Radio waves bounce all over the place and can get lost.)

So, if you're used to sending information to URLs (like **www.code.org**) and the servers actually have IP addresses for names (like 54.243.71.82) how does the Internet change from one to the other? That's what the DNS is for. The DNS (Domain Name Server) has tables that allow the system to go back and forth between URLs and IP addresses. If the Domain Name Servers ever stopped working, it would shut down the internet as we know it!

With that said, let's try to understand what the DNS does by making a little DNS table ourselves.

Lesson Tip

If you're thinking that this is a lot of text and it would be extremely boring to try to lecture this to a class full of elementary school kids, you're absolutely right! If you're unable to show a YouTube video in class to help explain it all, I highly recommend drawing pictures to explain each idea above, or choosing students as volunteers to act out what you describe while you're explaining. They're not expected to get every detail and definition at this point, only to gain exposure.

Pull out a piece of paper and draw a grid similar to that in the internet activity:

Sample of DNS Table:

#	URL	IP Address
1	code.org	54.243.71.82
2		
3		
4		
5		

First, we need to fill in this table.

- Survey the class for their favorite websites and write the URLs in the left column

- Use a site like **get-site-ip.com** to find the IP addresses for those sites and write them in the corresponding rows of the right column.

Now let's take this DNS Table and pretend to send messages through the internet!

Main Activity (20 min)

The Internet

Directions:

- Create your own DNS table, similar to what is shown above.
- Have the class help you fill in the blank spots in the table. Pick your favorite URLs and find their IP addresses using a site like **www.get-site-ip.com**.
- Divide into groups of 3 to 5.
- Assign each group an IP address from the newly created table, and assign each person in the group a position:
  - The Message Writer
  - The Internet
  - The Server (carries the IP address)
  - The Return Internet (optional)
  - The Message Receiver (optional)
- Each group will draw an **IP Address Cards and Delivery Type Cards - Manipulatives** to find out where their message is going and what their method of message delivery (Wi-Fi, Cable/DSL, or Fiber Optic Cable) will be.
- The Message Writer will craft a note to send to the server.
- The Internet will rip the message up into 4 small pieces called packets, then deliver each packet one at a time to the Server with the IP address that was drawn from the IP Address Card stack.
- The Server will make sure that the message arrives in order, then will send each packet off one at a time with the Return Internet (can be the same person or different person than the original Internet).
- The Return Internet will deliver each piece back to the Message Receiver (can be the same person or different person than the Message Writer) and put it back together.
- The Message Receiver will wait for all of the pieces to arrive, then read the message to be sure it arrived correctly!

Rules:

- The Internet must rip the message into exactly four packets.
- If the Internet drops a packet, they have to pick it up and go back to the start to deliver it again.
- The server has to wait for all of the message pieces to arrive before it can begin to send the message along.

Info:

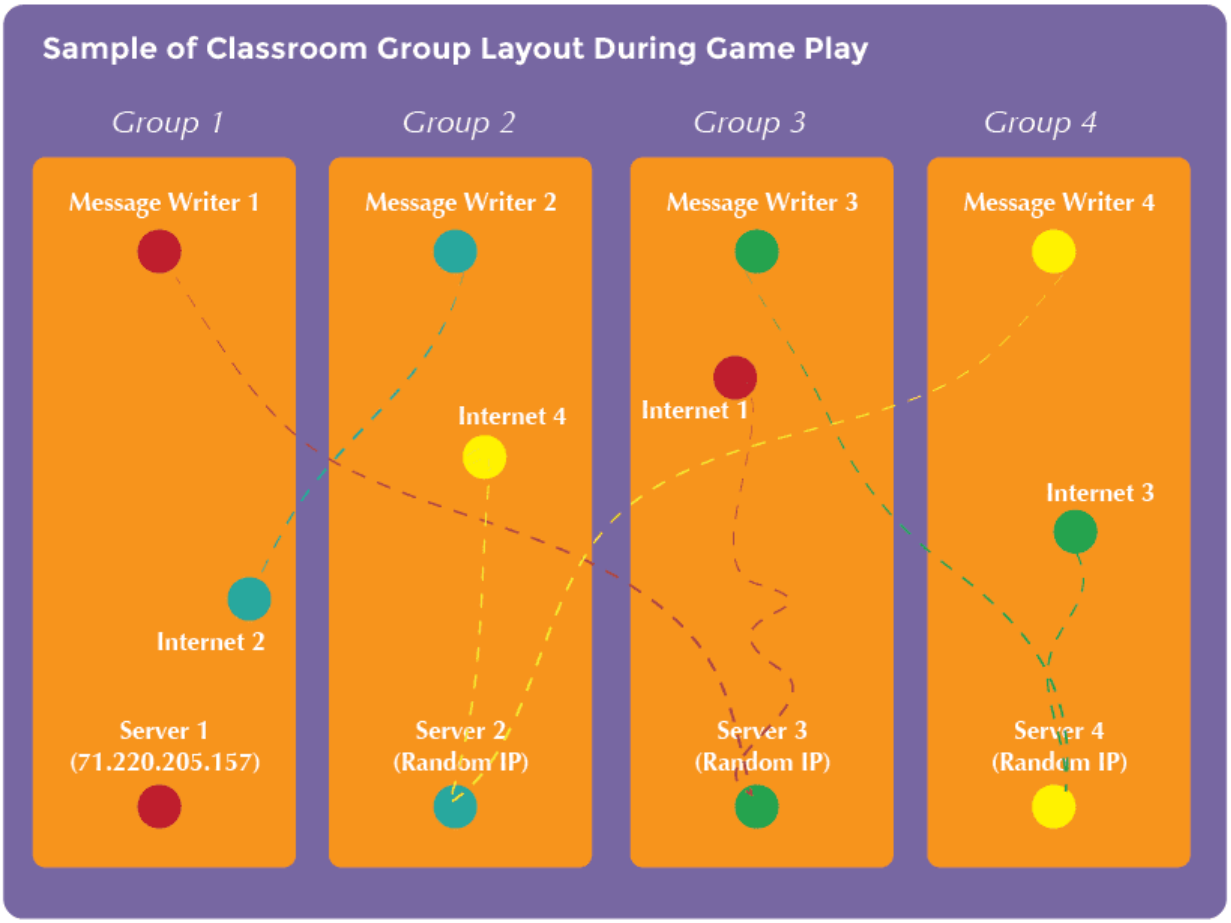
- Wi-Fi: Convenient, but spotty. Wi-Fi doesn't require cables, but since the signal bounces all over the place, packets can get lost pretty easily.
  - Simulation: Internet must carry each packet on their shoulder (no hands).
- Cable/DSL: Fairly good at delivering messages, but you must be connected to a wire.
  - Simulation: Internet must carry each packet on the back of one hand and must keep the other hand touching a wall, desk, chair or the floor at all times.
- Fiber Optic Cable: The best at delivering messages, but you must be connected to a wire.
  - Simulation: Internet can carry packets in hand, but must keep the other hand touching a wall, desk, chair or the floor at all times.

To play this game, you can have your groups cluster anywhere, but for the first time it can be less confusing to have groups play in a line.

Lesson Tip

If it feels like there are too many rules to explain outright, feel free to post them on the board and just explain the game as you go. You can play multiple rounds until the class really understands.

- Line up the "Servers" on one end of the room (holding their IP addresses). The Return Internet players can be over there as well (if you have that many people in each group).
- Have the everyone else line up across from their server at the other side of the room.
- The Message Senders will likely be sending their messages to a server other than their own, so the Internet players will likely cross over from group to group. It may look something like the diagram below (in English):



Wrap Up (15 min)

Flash Chat: What did we learn?

- What kind of connection would you rather have (Wi-Fi, DSL/Cable, or Fiber Optic)? Why?
- Why might it take your message a long time to get somewhere?

Lesson Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What's something you learned about the internet today?
- Why is learning about the internet important?

Assessment (5 min)

Internet - Assessment

Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.

Directions:

1) Create your own DNS table, similar to what is shown here.

2) Have the class help you fill in the blank spots in the table.
Pick your favorite URLs and find their IP addresses using a site like www.getip.com.

3) Divide into groups of 3 to 5.

4) Assign each group an IP address from the table, and each person in the group a position:

* The Message Writer
* The Internet
* The Server (carries the IP Address)
* The Return Internet (Optional)
* The Message Receiver (Optional)

5) Each group will draw an IP address Card and a Delivery Card to find out where their message is going and what their method of message delivery (Wi-Fi, Cable/DSL, or Fiber Optic Cable) will be.

6) The Message Writer will craft a note to send to the server.

7) The Internet will rip the message up into small pieces called packets, then deliver each packet one at a time to the Server with the IP address that was drawn from the IP address Card stack.

8) The Server will make sure that the message arrives in order, then will send each packet off one at a time with the Return Internet (can be the same person or different person than the original Internet).

9) The Return Internet will deliver each piece back to the Message Receiver (can be the same person or different person than the Message Writer) and put it back together.

10) The Message Receiver will wait for all of the pieces to arrive, then read the message to be sure it arrived correctly!

Rules:

1) The Internet must rip the message into exactly four packets.

2) If the Internet drops a packet, they have to pick it up and go back to the start to deliver it again.

3) The server has to wait for all of the message pieces to arrive before it can begin to send the message along.

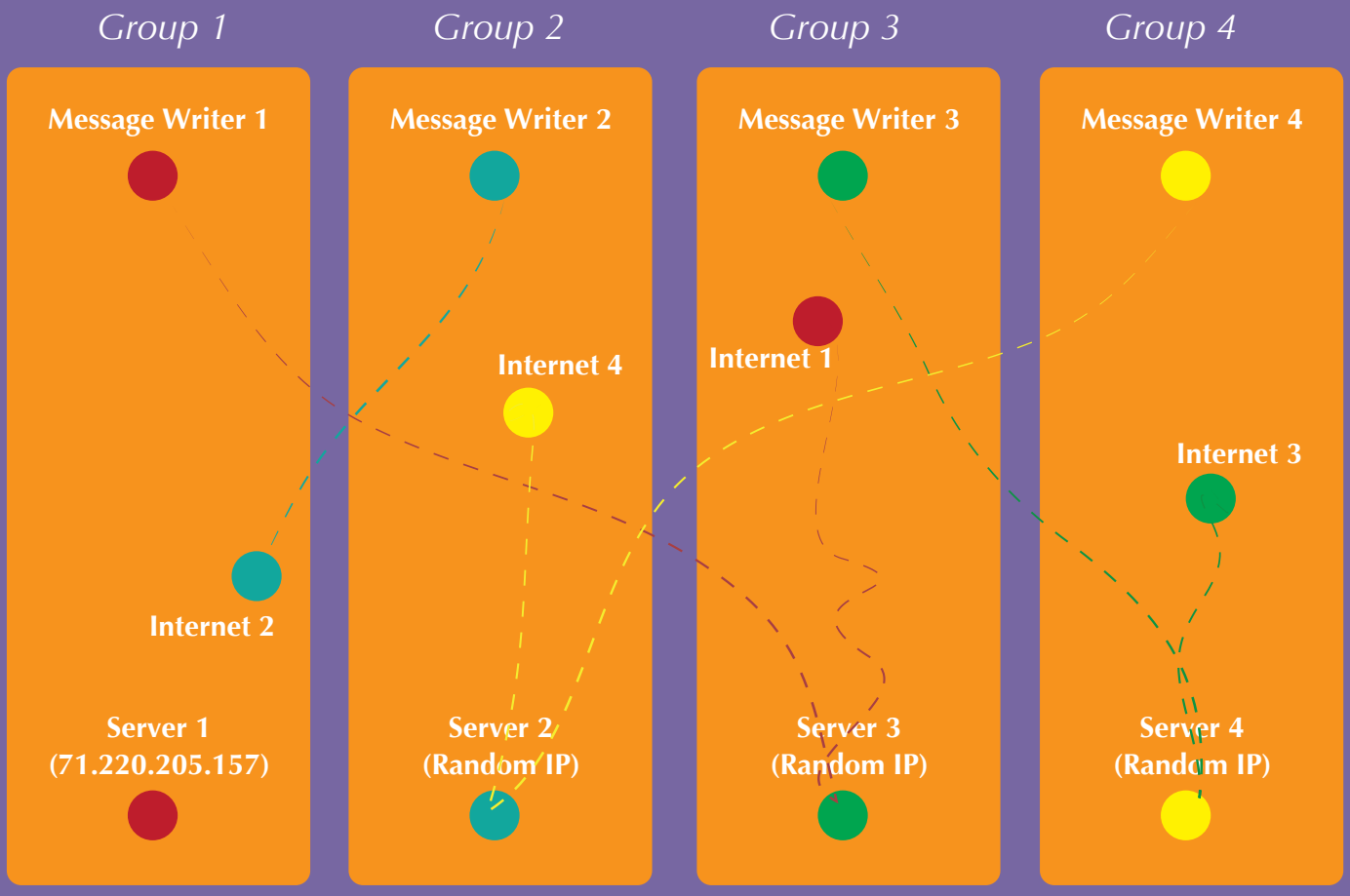
Internet

How the Internet Does what it Does

Sample of DNS Table

#	URL	IP ADDRESS
1	www.code.org	
2		
3		
4		
5		

Sample of Classroom Group Layout During Game Play



These cards correlate with numbered entries in the DNS Table.
(You should make one distinct row for each group.)

1	2
3	4
5	6

These cards correlate with different methods of delivering messages over the Internet.
(Print enough to have one card for each group.)

Wi-Fi	**Fiber Optic**
DSL	**Cable**

Types:

1) **Wi-Fi:** Convenient, but spotty. Wi-Fi doesn't require cables, but since the signal bounces all over the place, packets can get lost pretty easily.
Simulation: *Internet must carry each packet on their shoulder (no hands).*

2) **Cable/DSL:** Fairly good at delivering messages, but you must be connected to a wire.
Simulation: *Internet must carry each packet on the back of one hand and must keep the other hand touching a wall, desk, chair or the floor at all times.*

3) **Fiber Optic Cable:** The best at delivering messages, but you must be connected to a wire.
Simulation: *Internet can carry packets in hand, but must keep the other hand touching a wall, desk, chair or the floor at all times.*

Name: _____ Date: _____

U

Unplugged

The Internet

How the Internet Does What it Does

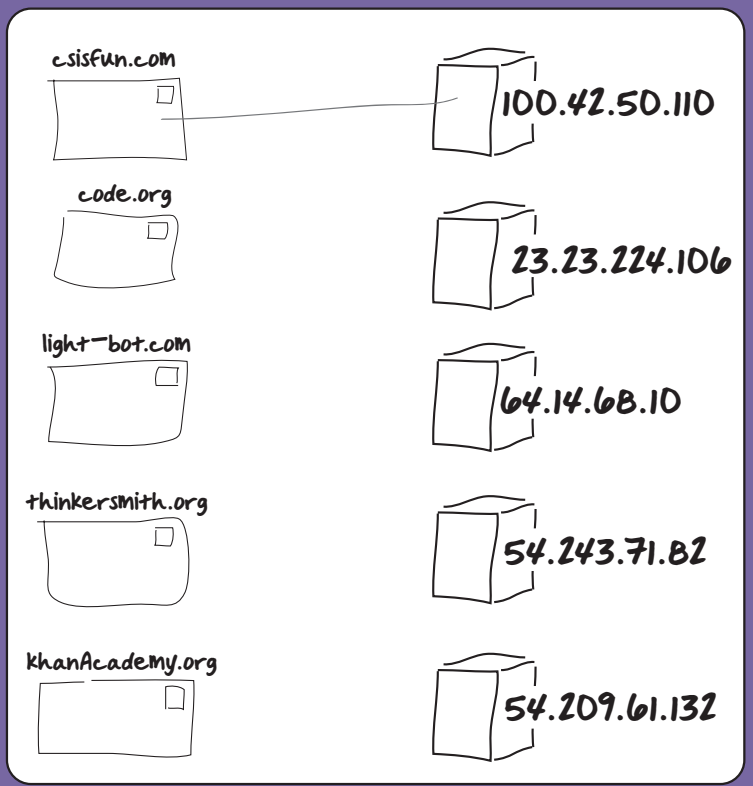
C O D E

The DNS has gone out, and now you're in charge of delivering information all over the Internet! Use the DNS Look-Up Table to figure out where each packet is supposed to go.

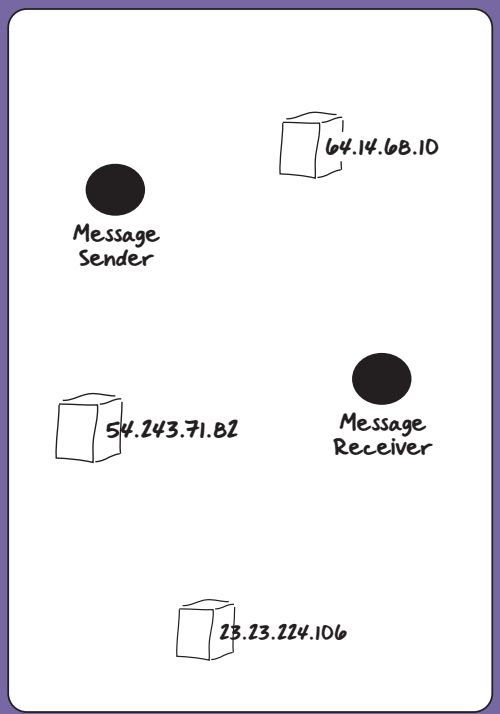
DNS Look-Up Table

#	URL	IP ADDRESS
1	code.org	54.243.71.82
2	csisfun.com	100.42.50.110
3	thinkersmith.org	64.14.68.10
4	light-bot.com	54.209.61.132
5	khanAcademy.org	23.23.224.106

Draw a line from each packet to the server where it is supposed to be delivered. The first one has been done for you.



This message is being delivered from someone at code.org to someone at thinkersmith.org. Draw the path that the message is likely to take.



Lesson 17: Crowdsourcing

Unplugged | Crowdsourcing

Overview

In computer science, we face some big, daunting problems. Challenges such as finding large prime numbers or sequencing DNA are almost impossible to do as an individual. Adding the power of others makes these tasks manageable. This lesson will show your students how helpful teamwork can be in the industry of computer science.

Purpose

It's very rare that one computer scientist works completely alone on a project. Even when that does happen, there is always benefit in numbers. Today, students will learn what it means to crowdsource a project. This activity builds teamwork and creates an efficient environment for students to solve problems.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Introduction**

Main Activity (20 min)

- **Crowdsourcing - Worksheet**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Extended Learning

Objectives

Students will be able to:

- Identify a large task that needs to be done.
- Rearrange a large task into several smaller tasks.
- Build a complete solution from several smaller solutions.

Preparation

- [ ] Watch the **Crowdsourcing - Teacher Video**.
- [ ] Review **Crowdsourcing - Worksheet**.
- [ ] Obtain a jar of lots of something (pennies, buttons, slips of paper, etc) and a deck of cards.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Crowdsourcing** - Teacher Video
- **Crowdsourcing** - Worksheet
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Crowdsourcing** - Getting help from a large group of people to finish something faster.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

Crowdsourcing - Say it with me: Crowd-sore-sing

Getting help from a large group of people to finish something faster.

Introduction

- Show your students your jar full of something.
  - "Look at this jar. I have a lot of buttons in here, and I need to tell the principal how many there are before the end of class."
  - "Can you think of a way I could get these counted quickly?"
- Your students may guide you toward seeking help, but if they don't, you can suggest it, too.
  - Pour all of the buttons (or pennies, etc.) into a pile on the floor.
  - Invite all of the students to come up and grab a small number (ten is good, but you can do more if your students can handle it).
  - Once they've counted out their ten, have them report to you, drop their buttons back in the jar, and go again until the pile is gone.
- Comment on how fast the task went.
  - Have the class reflect on how long it might have taken or how hard it may have felt to do alone.

Lesson Tip

Jars of buttons and pennies work nicely, but if you find yourself with little time to prepare, you can cut slips of paper and put them in a ziplock bag or even a pencil box.

Main Activity (20 min)

Crowdsourcing - Worksheet

Sometimes you have a big job that needs to get done, but it feels like it will take forever. Crowdsourcing is a way of using teamwork to make the job go much faster! In this game, we'll use crowdsourcing to sort decks of playing cards.

Directions:

1. Divide into groups of 4, 5, or 6.
2. Grab your deck of playing cards and dump it into a bag, bucket, or even a loose pocket that you can make with the bottom of your shirt.
3. Shake the cards until they're all mixed up.
4. Dump the cards out onto a table or desk where the whole group can see them.
5. Decide how to break up the task of sorting the deck so that every person has something to do and no one is doing too much.
6. Time yourself sorting the cards. Can you figure out a way to do it faster?
7. Repeat the game over and over until you think you have found the fastest way of crowdsourcing the card sorting activity.

Lesson Tip

It can be challenging for students to figure out how to break apart large tasks at first. Students might find it helpful to have some ideas handed to them after working for a while. One great division for sorting cards is as follows:

- One person picks up the cards and determines the suit of each one.
- One person manages Hearts.
- One person manages Diamonds.
- One person manages Clubs.
- One person manages Spades.
- (If there's another, they can put all sorted suits back together again.)

Wrap Up (15 min)

Flash Chat: What did we learn?

- Have you ever tried to sort a pile of cards by yourself?
- Do you think it was easier or harder to have help?
- What other things do you have to do in life that could be easier with help?

Lesson Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What are the benefits of crowdsourcing?
- What kind of things do you want to make with computer science? How do you see crowdsourcing being beneficial in those projects?

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Reverse Crowdsourcing

Often we think of crowdsourcing as pulling things apart to make them more simple. You can also make big, beautiful things with the same technique.

Have your students each grab three cards and build one segment of a **card house**. Each student can go one after another to build a grand card tower.

Try with two, or even three students adding their chunk at a time.

- Does crowdsourcing always make a task easier?

Crowdsourcing in the Round

- You can crowdsource all at the same time or you can do it one person at a time. Try having the whole class sort the same deck of cards, one student at a time.
  - Shuffle the cards and place them in a pile in the center of the room.
  - Have each student approach the pile and choose four cards.
    - Have four piles for the students to sort their cards into
      - Spades
      - Clubs
      - Hearts
      - Diamonds
    - Once all cards have been put in their four piles, have the following four students sort the individual piles.
    - The last person will put all four piles together.
- This version may not save a lot of time, but it still divides the work and lets each individual have more free time!

U

Unplugged

Name: _____ Date: _____

Crowdsourcing

Working together to make big things happen

C O D E

Sometimes you have a big job that needs to get done, but it feels like it will take forever. Crowdsourcing is a way of using teamwork to make the job go much faster!

In this game, we'll use crowdsourcing to sort decks of playing cards.

Directions:

1) Divide into groups of 4, 5, or 6.

2) Grab your deck of playing cards and dump it into a bag, bucket, or even a loose pocket that you can make with the bottom of your shirt.

3) Shake the cards until they're all mixed up.

4) Dump the cards out onto a table or desk where the whole group can see them.

5) Decide how to break up the task of sorting the deck so that every person has something to do and no one is doing too much.

6) Time yourself sorting the cards. Can you figure out a way to do it faster?

7) Repeat the game over and over until you think you have found the fastest way of crowdsourcing the card sorting activity.

New Word!

Crowdsourcing

Say it with me: Crowd-sore-sing

Getting help from a large group of people to finish something faster

It would be easier to clean your room if you tried **crowdsourcing** the work with a bunch of friends.

Course F

Lesson 1: Algorithms Unplugged: Tangrams

Unplugged | Algorithms

Overview

This lesson shows us something important about algorithms. As long as you keep an algorithm simple, there are lots of ways to use it. However, if you want to make sure everyone produces the same outcome, then your algorithm needs more detail. Students will learn the difference between a detailed and general algorithm while playing with tangrams.

Purpose

By introducing a basic concept like *algorithms* to the class in an unplugged activity, students who are intimidated by computers can still build a foundation of understanding on these topics. Algorithms are essential to computer science. In this lesson, students will learn how to translate instructions into a algorithm and how that plays a role in programming.

Agenda

Warm Up (10 min)
- **Vocabulary**
- **Introduction**

Main Activity (20 min)
- **Algorithms**

Wrap Up (15 min)
- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (10 min)
- **Algorithms Unplugged: Tangram - Assessment**

Objectives

Students will be able to:

- Tackle the challenge of translating an image into actionable instructions.
- Convey instructions to teammates in order to reproduce an image.
- Analyze the work of teammates to determine whether an outcome was successful.

Preparation

- [ ] Watch the **Algorithms Unplugged: Tangrams - Teacher Video**.
- [ ] Give every student a **Think Spot Journal**.
- [ ] Print out a **Tangram Set & Algorithm Card Images Pack - Manipulatives** for every student.
- [ ] Print out a **Algorithms Unplugged: Tangram - Assessment** for every student.

Links

For the Teacher

- **Algorithms Unplugged: Tangrams** - Teacher Video
- **Tangram Set & Algorithm Card Images Pack** - Manipulatives
- **Algorithms Unplugged: Tangram** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Algorithm** - A list of steps to finish a task.

Teaching Guide

Warm Up (10 min)

Vocabulary

- **Algorithm** - Say it with me: Al - gor - ith - him

A list of steps to finish a task.

Introduction

Lesson Tip

If your class has never used tangram pieces, you can choose to do an example for them or even have an entire tangram lesson. There are several good ones on the Internet. **Here** is a lesson that you can do in the classroom and **here** is a game that you can play online.

Your students may or may not have played with tangrams before. If they have, you can skip this portion, and move right to explaining the main activity.

Explain to the students that tangrams are usually used to solve puzzles. You receive a set of seven "tans" and must use them all (without overlapping any) to recreate an image that has been given to you. Often, this is done as an individual activity, and the player is allowed to see the image that they are trying to recreate. Many times, you can lay your pieces right on top of the image silhouette to be sure that the solution is just right.

Main Activity (20 min)

Algorithms

We are going to use our tangrams in a slightly different way than most. Instead of looking at our puzzles and trying to guess which shape goes where, we are going to get puzzles that already tell you where each shape goes.

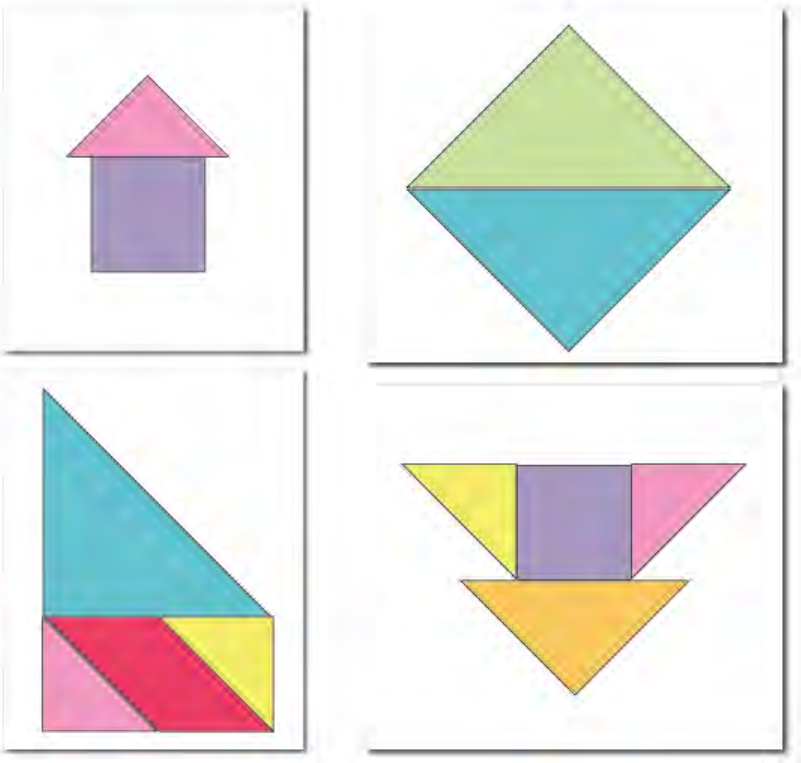
You might think that this will make it easier, but it won't, because students will also not get to actually look at the image that we are trying to recreate! Instead, a teammate will be describing the image to us.

To keep it from getting too difficult, we will not use puzzles that require all seven pieces.

Directions:

1. Divide into groups of 3-5.
2. Each player should cut out their own set of tangrams.
3. Have one member of each group select an Algorithm Card without showing it to anyone else.
4. The person with the Algorithm Card will try to explain the image to everyone else without letting them actually see it.
5. The other players will build their pictures off of the description given by the Card Holder.
6. When the Card Holder is done, everyone will show their pictures and see if they all ended up with the same image.
7. If everyone ends up with the same drawing, the Card Holder can show the card and see if everyone matched the card.
8. If any of the pictures in the group are different from each other, have the Card Holder try describing the image again, using more detail.
9. Choose a new Card Holder and a new Algorithm Card and repeat until everyone has had a chance to describe an image.

Play through this several times, with images of increasing difficulty.



Wrap Up (15 min)

Flash Chat: What did we learn?

- What did we learn today?
- Was it easier or harder than you thought it would be to describe an image to one another?
- Did any group end up having arrangements that all matched?
- Can you share some tricks that you came up with that helped your group match the Image Card exactly?

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What did you learn today?
- How do you feel about today's lesson?
- Can you think of tricks to make it easier to describe tangram pictures to a partner?
- Describe why you might want to be very detailed when creating algorithms for writing code.

Assessment (10 min)

Algorithms Unplugged: Tangram - Assessment

Pass out the assessment and allow time for students to complete it. If there is extra time, go over the answers as a class.

Name: _____ Date: _____

Algorithms

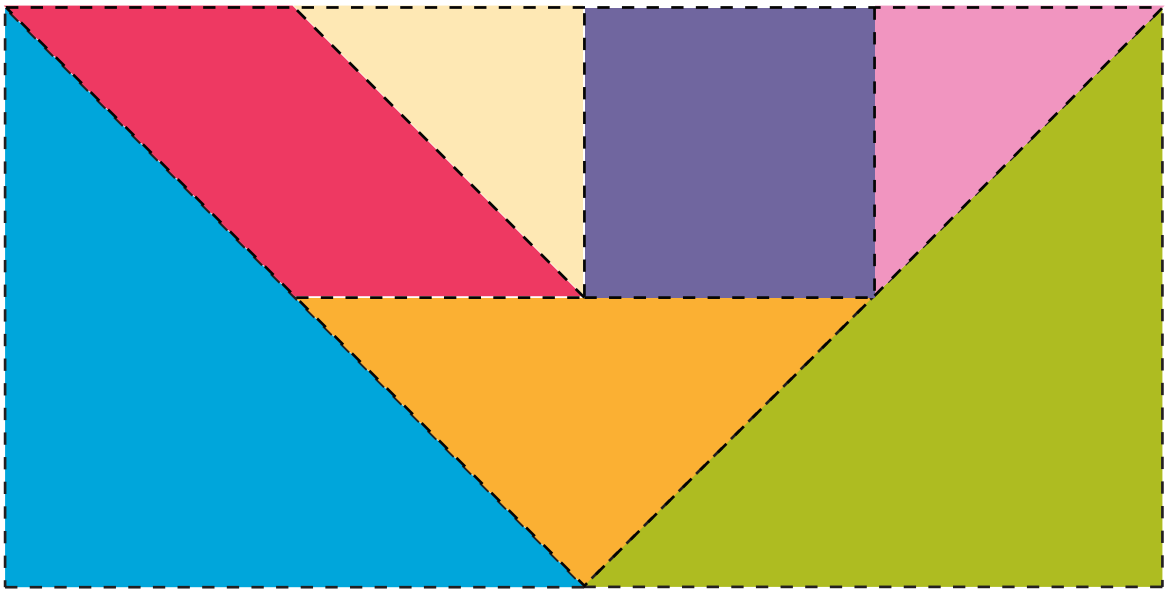
Tangrams Algorithm Activity

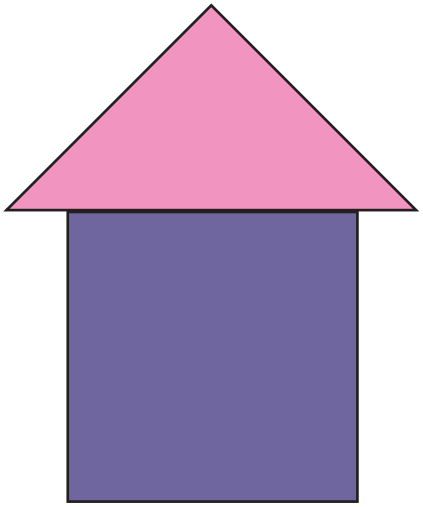
This lesson shows us something important about algorithms. If you keep an algorithm simple there are lots of ways to use it. If you want to make sure everyone ends up with the same thing, then your algorithm needs to have a lot of detail.

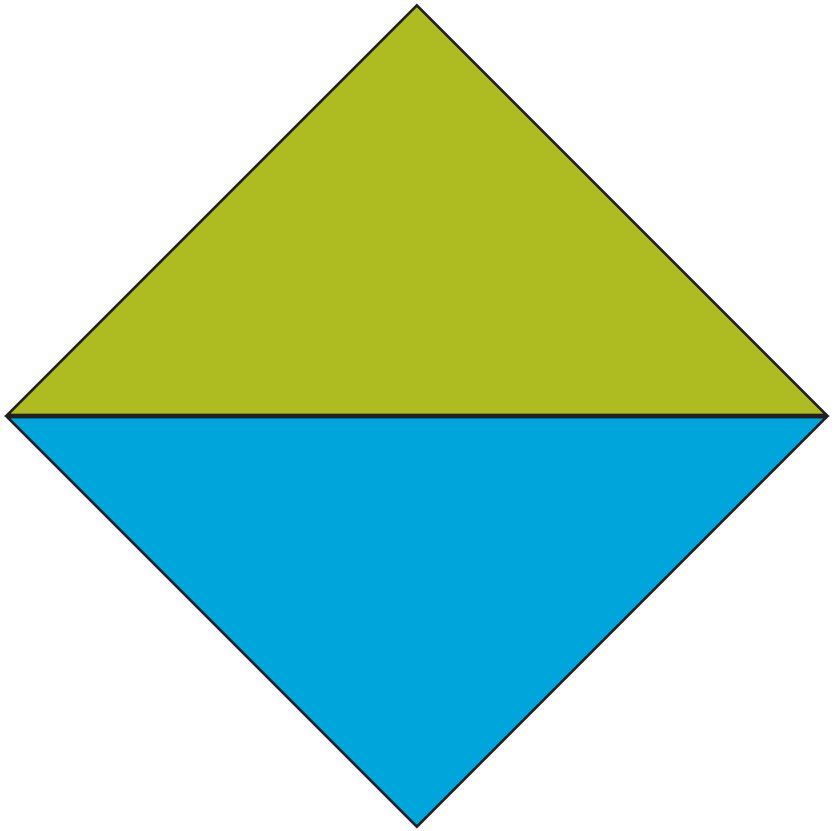
This activity will show both options.

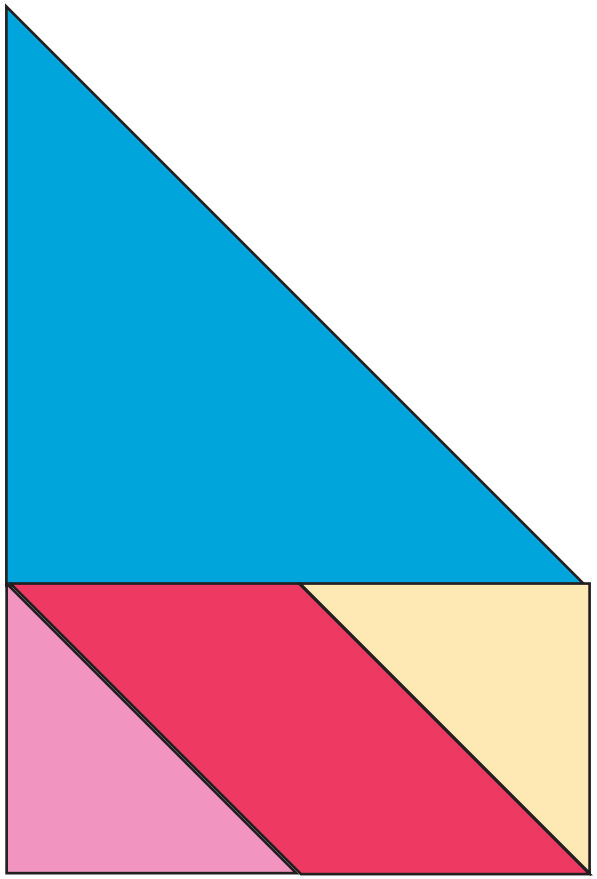
Directions:

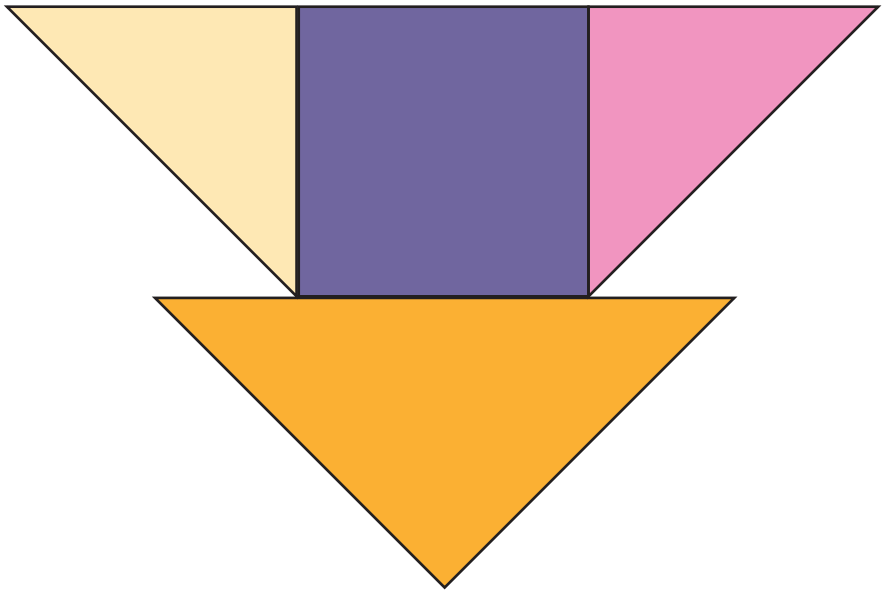
1. Divide into groups of 3-5.
2. Each player should cut out their own set of tangrams.
2. Have one member of each group select an Algorithm Card without showing it to anyone else.
3. The person with the Algorithm Card will try to explain the image to everyone else without letting them actually see it.
4. The other players will build their pictures off of the description given by the Card Holder.
5. When the Card Holder is done, everyone will show their pictures and see if they all ended up with the same image.
6. If everyone ends up with the same drawing, the Card Holder can show the card and see if everyone matched the card.
7. If any of the pictures in the group are different from each other, have the Card Holder try describing the image again, using more detail.
8. Choose a new Card Holder and a new Algorithm Card and repeat until everyone has had a chance to describe an image.











Name: _____ Date: _____

Algorithms

Tangrams Assessment Worksheet

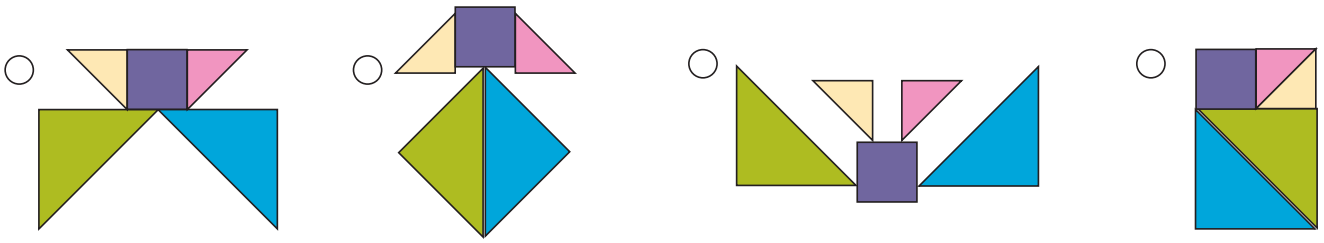
Very specific algorithms help multiple people create identical products.

Less specific algorithms allow a great deal of flexibility for every person to have something different.

Circle the drawing that does not follow the algorithm provided.

Algorithm #1

1) Put two large triangles at the bottom of the image.
2) Put a square on top of those two triangles.
3) Put two little triangles beside the square.



Circle the algorithm that goes with Drawing 1.

Algorithm A

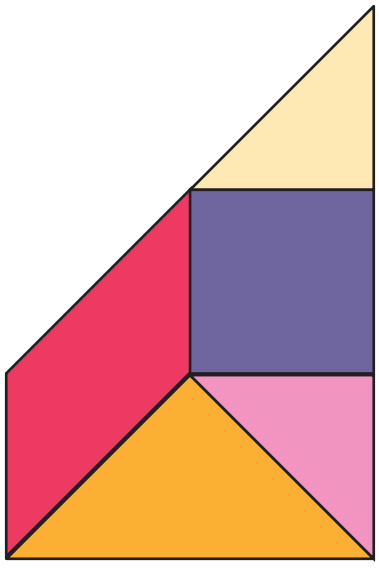
1) Use two triangles, a square, and another piece
2) Line two triangles up with the square
3) Put the last piece on top of the square

Algorithm B

1) Use three triangles, a rhombus, and another piece
2) Put the rhombus at the bottom
3) Put all three triangles above the rhombus
4) Put the final piece to the left of everything else

Algorithm C

1) Use three triangles, a square, and another piece
2) Line two triangles up with the square
3) Put a third triangle beneath the other shapes
4) Put the last piece on the left



Drawing 1

Lesson 3: Common Sense Education: The Power of Words

Common Sense Education | Cyberbullying

Overview

Students consider that while they are enjoying their favorite websites they may encounter messages from other kids that can make them feel angry, hurt, sad, or fearful. They explore ways to handle cyberbullying and how to respond in the face of upsetting language online.

Students discuss all the ways they use technology for communication and explore the similarities and differences between in-person and online communication. Students then brainstorm ways to respond to cyberbullying.

Purpose

This lesson will provide students with the tools that they need to handle cyberbullying if they are ever in the situation of having someone negatively responds to their online postings.

Students may not ever have the misfortune of experiencing cyberbullying, but they should understand what it is so that they can spot it online. Students will learn how to identify cyberbullying and what steps they should take to make it stop. This may become helpful in later puzzles when students have the opportunity to share their work.

Agenda

Warm Up (5 min)
- **Introduction**

Main Activity (35 min)
- **What's the Problem?**
- **Crossing the Line**
- **Talk and Take Action**

Wrap Up (15 min)
- **Flash Chat: What did we learn today?**
- **Journaling**

Assessment (10 min)

Objectives

Students will be able to:

- Empathize with those who have received mean and hurtful messages.
- Judge what it means to cross the line from harmless to harmful communication online.
- Generate solutions for dealing with cyberbullying.

Preparation

- [ ] Preview the **Common Sense Education - Power of Words - Teacher Prep Guide** and prepare to show it to your class.
- [ ] Print out the **Words Can Hurt Handout** from **Common Sense Education - Power of Words - Teacher Prep Guide** (page 7) for each group of four.
- [ ] Print out the **Talk and Take Action Handout** from **Common Sense Education - Power of Words - Teacher Prep Guide** (page 6) for each student.
- [ ] Print out the assessment on page 8-9 of **Common Sense Education - Power of Words - Teacher Prep Guide**.
- [ ] Obtain colored pencils and a string the length of the classroom.

Links

For the Teacher

- **Common Sense Education - Power of Words** - Teacher Prep Guide
- **The Power of Words** - Lesson Video
- **Common Sense Education** - Website
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Cyberbully** - Using technology tools to deliberately upset someone else.

Teaching Guide

Warm Up (5 min)

Introduction

Draw a series of expressive faces on the board. View **Feeling Faces** for examples.

Invite the students to suggest emotions that match each face's expression. With every suggestion, write the emotion next to the feeling face. Answers will vary.

Tell students that not everyone will react to a particular situation in the same way, but just because a reaction is different from our own, doesn't mean we should discount others' feelings.

Explain to students they are going to watch a video about how words, whether typed or spoken, can impact how someone else feels.

Show students **The Power of Words - Lesson Video**.

Ask:

- Who has heard of the saying, "Sticks and stones may break my bones, but words will never hurt me"?
- What did Guts mean in his text that sometimes words can hurt?
  - Words are powerful. Sometimes it is hard to ignore what someone is saying when it's a mean name. Names *can* make you feel sad or hurt.

Remind students to keep Leg's question in the back of their mind during this lesson: *How do you treat others online?*

Main Activity (35 min)

What's the Problem?

Organize students into groups of four and have each group pick a person to record their ideas.

Distribute the **Words Can Hurt Student Handout**. Have the groups of students read the scenario about Rani and Aruna receiving mean messages through a children's game website.

Have each group answer the questions, then have them share their responses with the class. Look for responses that show empahty for Rani and Aruna and acknowledge that the messages sent to them were mean and hurtful. Ask the students to read the 'Use Common Sense!' section on the **Words Can Hurt Student Handout**.

Invite students to share their own stories.

Ask:

- Have you seen mean messages sent to you or others online? Tell us about it, but do not use real names.

Divide students into pairs.

Invite one partner to write the phrase "You're weird" on a piece of paper, then hand it to their partner. Tell them that they just received this text.

Ask:

- What are the reasons the person might have texted "You're weird"?
  - They're continuing an inside joke; the first person did something silly at an earlier time; a group of kids is teasing the kid; the person who sent the text really does think the person is weird but is afraid to say it to his or her face.
- How did the partner feel about being called weird?
  - Possibly like the other person was kidding around, but maybe that the person was teasing or being hurtful.

Tell one person from each pair to say to the other person, "You're weird," with a smile on their face.

Ask:

- Why might you feel differently if you could see the person?
  - People give non-verbal cues through facial expressions and body language.

Crossing the Line

Place the piece of string across the length of the classroom. Ask students to stand on one side of the line. Then ask them to imagine that they are online and somebody has sent them a message, which you will read to them. Tell the students tot stay where they are if they think the message is okay; to cross over the line if they think the message is not okay; or to stand on the line if they think the message is in between.

Read:

- You are my friend.
- You are an idiot.
- I'm having a party and you're not invited.
- I like your new haircut.
- You are ugly.
- Thanks for the advice. Next time, will you tell me in person rather than through text?
- Did you finish your homework?
- Why is it taking you so long to finish it?
- You are such a freak.

Review with the students that kids like to go online and use cell phones to email, chat, watch videos, send messages, play games, and do homework. But sometimes the language can get mean or scary. Messages that make people feel bad cross the line. Sometimes that meanness is unintentional, but when people use tools such as the internet and cell phones to deliberately upset someone else over and over, that's cyberbullying.

Talk and Take Action

Have students return to their seats.

Discuss how easy is it to feel angry or upset when somebody sends you a mean or scary message online.

Define:

- **Cyberbullying**: Using technology tools such as the internet and cell phones to deliberately upset someone else.

Explain that cyberbullies deliberately to try to make you feel that way, just like real-life bullies.

Discuss:

- Cooling down can be good first step when you receive a mean message online. Taking a deep breath, counting backwards from 10, or pausing to think about what you will do next can give you time to think of the BEST way to handle the situation.
- Finding help or telling a trusted adult or friend can be a good way to take action. You shouldn't deal with the cyberbullying situation alone. The person you tell should be someone who wants to hear what you have to say and will help you work on a solution. Adults can be especially good because they often have the power to influence the situation or they can give you advice about what to do.
- Ignoring the person who is cyberbullying you can be very effective. Those who bully often like attention.
- Whatever you do, remember to keep a copy of your communication with the individual who is cyberbullying you. f you delete the communication, there is no proof of how the bully treated you if you need to show a trusted adult.

Distribute the **Talk and Take Action Student Handout** to each student. Encourage them to depict a cyberbullying scenario and possible solution. They can use pencils and paper to make the comics.

Wrap Up (15 min)

Flash Chat: What did we learn today?

You can use these questions to assess your students' understanding of the lesson objectives. You may want to ask students to reflect in writing on these questions in their **Think Spot Journal**.

Ask:

- Why is it a bad idea to send mean or scary messages online?
  - Because they can make the person who gets the message upset, angry, or scared.
- Why might there be more misunderstandings between people when they send online messages as opposed to a face-to-face discussion?
  - Online messages can be more confusing or scarier than face-to-face messages because there are no face-to-face cues to help you understand people's intentions.
- What can kids do when they get cyberbullying messages?
  1. Stay calm and take a deep breath
  2. Tell a friend or trusted adult who can help develop a plan to handle the situation
  3. Ignore the bully
  4. Keep a copy of the communication with the bully.

Journaling

Having students write about what they learned, why it’s useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is cyberbullying?
- Who are some people you can go to if you are ever bullied online or in person?

Assessment (10 min)

Print out the assessment from **Common Sense Education - Power of Words - Teacher Prep Guide** (page 8-9) and distribute it to the class. Give students enough time to complete the assessment, but make sure there is enough time to go over answers.

The Power of Words

Essential Question

What should you do when someone uses mean or scary language on the Internet?

Lesson Overview

Students consider that while they are enjoying their favorite websites they may encounter messages from other kids that can make them feel angry, hurt, sad, or fearful. They explore ways to handle cyberbullying and how to respond in the face of upsetting language online.

Students discuss all the ways they use technology for communication, put themselves in the shoes of children who are cyberbullied on a kids' game website, and explore both the similarities and differences between in-person versus online communication. Students then brainstorm ways to respond to cyberbullying.

Learning Objectives

Students will be able to ...

- empathize with those who have received mean and hurtful messages.
- judge what it means to cross the line from harmless to harmful communication online.
- generate solutions for dealing with cyberbullying.

Materials and Preparation

- Preview the video "**The Power of Words**," and prepare to show it to students.
- Copy the **Words Can Hurt Student Handout,** one for every four students.
- Copy the **Talk and Take Action Student Handout,** one for each student.
- Colored pencils
- String (cut string the length of the classroom)

Family Resources

- Send home the **Cyberbullying Family Tip Sheet (Elementary School)**.

Estimated time: 45 minutes

Standards Alignment –

Common Core:

grade 3: RI.1, RI.3, RI.4, RI.10, RF.4a, W.4, SL.1a, SL.1b, SL.1c, SL.1d. SL.3, SL.4, SL.6, L.3a, L.6

grade 4: RL.3, RL.10, RI.1, RI.3, RI.4, RI.7, RI.10, RF.4a, W.9b, SL.1a, SL.1b, SL.1c, SL.1d, SL.4, SL.5, L.3a, L.6

grade 5: RL.3, RL.10, RI.1, RI.3, RI.4, RI.7, RI.10, RF.4a, W.9b, SL.1a, SL.1b, SL.1c, SL.1d, SL.4, SL.5, SL.6, L.6

NETS•S: 2b, 5a, 5d

Key Vocabulary –

cyberbully (verb): using technology tools such as the Internet and cell phones to deliberately upset someone else

introduction

Warm-up *(5 minutes)*

DRAW a series of expressive faces, or emojis, on the board: 

INVITE students to suggest emotions that match each face's expression. Answers will vary.

Faces	Responses will vary
	happy, glad, excited, content, thrilled, pleased
	mad, angry, frustrated, grouchy, furious, upset
	bored, calm, relaxed, lonely, disinterested (note: this face may evoke opposing emotions as it is pretty neutral)
	sad, unhappy, upset, depressed, miserable
	surprised, scared, shocked, amazed

TELL students that not everyone will react to a particular situation the same way, but just because a reaction is different from our own, that doesn't mean we should discount others' feelings.

EXPLAIN to students they are going to watch a video about how words, whether typed or spoken, can impact how someone else feels.

SHOW students "**The Power of Words**."

ASK:

• *Who has heard of the saying, "Sticks and stones may break my bones, but words will never hurt me"?*	Responses will vary.
• *What did Guts mean in his text that sometimes words can hurt?*	Despite the old saying, "*Sticks and stones may break my bones, but words will never hurt me*," words are powerful. Sometimes it is hard to ignore what someone is saying when it's a mean name. Names CAN make you feel sad or hurt.

REMIND students to keep Legs's question in the back of their minds during this lesson: *How do you treat others online?*

teach 1

What's the Problem? *(15 minutes)*

ORGANIZE students into groups of four, and have each group pick a person to record their ideas.

DISTRIBUTE the **Words Can Hurt Student Handout**. Have the groups of students read the scenario about Rani and Aruna receiving mean messages through a children's game website.

HAVE each group answer the questions, and then have them share their responses with the class. Look for responses that show empathy for Rani and Aruna and acknowledge that the messages are mean and hurtful and should be stopped. Ask students to read the Use Common Sense! section on the **Words Can Hurt Student Handout**.

INVITE students to share their own stories.

ASK:

Have you seen mean messages sent to you or others online? Tell us about it, but do not use real names.	Answers will vary.

DIVIDE students into pairs.

INVITE one partner to write the phrase "You're weird" on a piece of paper, and then hand it to their partner. Tell them that they just received this text.

ASK:

What are the reasons the person might have texted "You're weird"?	They're continuing an inside joke; the first person did something silly at an earlier time; a group of kids is teasing the kid; the person who sent the text really does think the person is weird but is afraid to say it to his or her face.
How did the partner feel who was called weird?	Possibly like the other person was kidding around, but maybe that the person was teasing or being hurtful.

TELL one person from each pair to say to the other person, "You're weird," with a smile on his or her face.

ASK:

Why might you feel differently if you could see the person?	People give non-verbal cues through facial expressions and body language.

teach 2

Crossing the Line *(10 minutes)*

PLACE the piece of string across the length of the classroom. Ask students to stand on one side of the line. Then ask them to imagine that they are online and somebody has sent them a message, which you will read to them. Tell the students to stay where they are if they think the message is okay; to cross over the line if they think the message is not okay; or to stand on the line if they think the message is in between.

READ each of these messages aloud and have students move accordingly:

- *You are an idiot.*
- *I'm having a party and you're not invited.*
- *I like your new haircut.*
- *You are really ugly.*
- *Thanks for the advice. Next time would you mind telling me in person rather than by texting?*
- *Did you finish your homework?*
- *Why is it taking you so long to finish it?*
- *You are such a freak.*

REVIEW with students that kids like to go online and use cell phones to email, chat, watch videos, send messages, play games, and do homework. But sometimes the language can get mean or scary. Messages that make people feel bad cross the line. Sometimes that meanness is unintentional, but when people use tools such as the Internet and cell phones to deliberately upset someone else over and over, that's cyberbullying.

teach 3

Talk and Take Action *(10 minutes)*

HAVE students return to their seats.

DISCUSS how easy it is to feel angry or upset when somebody sends you a mean or scary message online.

DEFINE the Key Vocabulary term **cyberbullying**. Explain that cyberbullies deliberately try to make you feel that way, just like real-life bullies. Discuss the following ideas about what they can do when faced with cyberbullying:

- *Cooling down can be a good first step when you receive a mean message online. Taking a deep breath, counting backwards from 10, or pausing to think about what you will do next can give you time to think of the BEST way to handle the situation.*
- *Finding help or telling a trusted adult or a friend can be a good way to take action. You shouldn't deal with the cyberbullying situation alone. The person you tell should be someone who wants to hear what you have to say, and will help you work on a solution. Adults can be especially good because they often have the power to influence the situation, or can give you advice about what to do.*
- *Ignoring the person who is cyberbullying you can be very effective. Those who bully often like attention.*
- *Whatever you do, remember to keep a copy of your communication with the individual who is cyberbullying you. If you delete the communication, there is no proof of how the bully treated you if you need to show it to a trusted adult.*

DISTRIBUTE the **Talk and Take Action Student Handout** to each student. Encourage them to depict a cyberbullying scenario and a possible solution. They can use pencils and paper or go online and use the free tool Make Beliefs Comix (www.makebeliefscomix.com) to complete a comic strip.

closing

Wrap-up *(5 minutes)*

You can use these questions to assess your students' understanding of the lesson objectives. You may want to ask students to reflect in writing on one of the questions, using a journal or an online blog/wiki.

ASK:

Why is it a bad idea to send mean or scary messages online?	Because they can make the person who gets them upset, angry, or scared.
Why might there be more misunderstandings between people when they send online messages as opposed to face-to-face discussion?	Online messages can be more confusing or scarier than face-to-face messages because there are no face-to-face cues to help you understand people's intentions.
What can kids do when they get cyberbullying messages?	They can 1) calm down and take a deep breath, 2) tell a friend or a trusted adult who can help develop a plan to handle the situation, 3) ignore the bully, 4) keep a copy of the communication with the bully.

The Power of Words

Directions

Create a cartoon about a cyberbullying situation. Each frame should show a different part of the situation:

FRAME 1: Make a cartoon about something that a cyberbully might do or write online. Remember to use language appropriate for school.

FRAME 2: Show what you might do if you saw what the cyberbully has done or written.

FRAME 3: What might be a positive outcome, or result, of the situation?

What might a cyberbully say or do?	What would you do in response?
What would a positive outcome be?	

Use Common Sense!

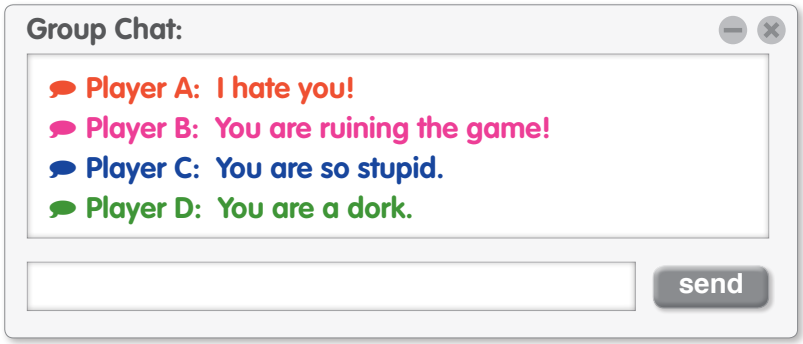
- If someone is mean to you online, take a breather and ignore them. Save a copy of your conversation between you and the bully.
- If you, or someone you know, is cyberbullied, talk to a trusted adult – like a parent, family member, or teacher. Together, you can think of a plan for how to respond.

The Power of Words

Directions

Read the story below and then answer the questions that follow.

Rani and Aruna love a website that has games and chatting for kids. Their parents let them play on the site. Lately, though, Rani and Aruna have been receiving mean messages on the site, including:



1. How do you think Rani and Aruna feel when they read those messages?

Rani and Aruna feel _____

_____

2. How would you feel if you received messages like these?

I would feel _____

_____

3. Why do you think people send these kinds of message to people they don't know?

People send these kinds of messages because

_____

Use Common Sense!

There's an old saying: "Sticks and stones may break my bones, but words will never hurt me."

I think that this saying is TRUE/NOT TRUE (circle one) because

_____

The Power of Words

1. Alicia receives a text message from her friend Ronald. The message says, "I am having a party. You are not invited." Circle the word that shows how Alicia might feel after she receives the message.

a) hurt

b) excited

c) popular

2. Theo is having fun playing Dragons and Knights online. Then he sees a message from another player. It says, "You're ruining the game, stupid!" What should Theo do about the message? What shouldn't he do? Use the chart below to fill in the letters that go with each answer.

a) Ignore the player who sent the mean message

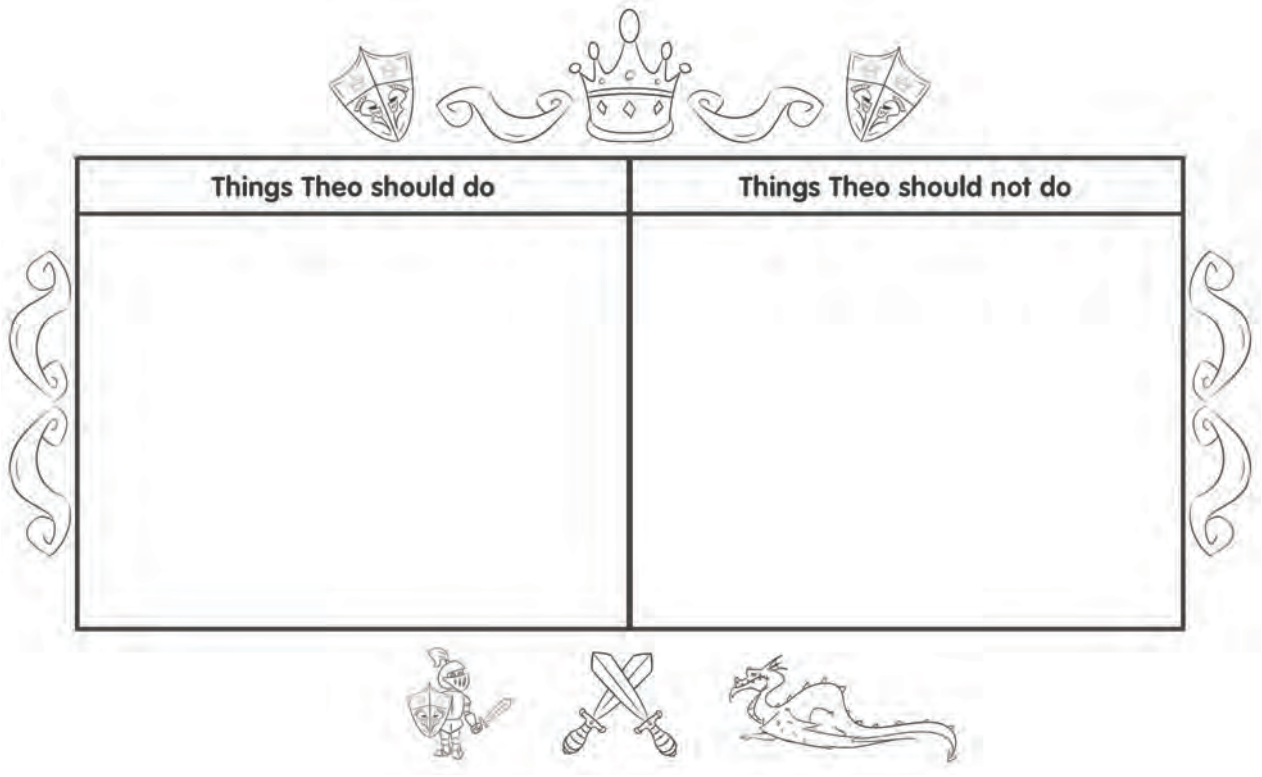
b) Write a message back that says, "You're so stupid, you're the one ruining the game"

c) Pretend that he doesn't feel hurt by the message

d) Save the message in case the other player sends Theo another mean message

e) Tell an adult about the message

f) Never play Dragons and Knights online again



Things Theo should do	Things Theo should not do

The Power of Words

3. The following acronym, STOP, gives advice on what to do when something goes wrong online. Explain what each letter means.

STOP	What is the meaning of each phrase? Explain in your own words.
Step away	
Tell a trusted adult	
Okay sites first	
Pause and think online	

The Power of Words

1. Alicia receives a text message from her friend Ronald. The message says, "I am having a party. You are not invited." Circle the word that shows how Alicia might feel after she receives the message.

a) hurt

b) excited

c) popular

Answer feedback

The correct answer is **a**. Alicia probably feels **hurt** by Ronald's message. Telling someone they aren't invited can hurt their feelings. Things that are hurtful in person are also hurtful online.

2. Theo is having fun playing Dragons and Knights online. Then he sees a message from another player. It says, "You're ruining the game, stupid!" What should Theo do about the message? What shouldn't he do? Use the chart below to fill in the letters that go with each answer.

Answer feedback

Things Theo should do	Things Theo should not do
a) Ignore the player who sent the mean message d) Save the message in case the other player sends Theo another mean message e) Tell an adult about the message	b) Write a message back that says, "You're so stupid, you're the one ruining the game" c) Pretend that he doesn't feel hurt by the message f) Never play Dragons and Knights online again

The Power of Words

3. The following acronym, STOP, gives advice on what to do when something goes wrong online. Explain what each letter means.

STOP	What is the meaning of each phrase? Explain in your own words.
Step away	When something goes wrong, you should step away from the device or website and take a break.
Tell a trusted adult	If something goes wrong, you should tell a trusted adult so that he or she can help you.
Okay sites first	Make sure you are visiting sites that are appropriate for your age.
Pause and think online	When you go online, you should pause and think twice before sending, posting, or reacting.

Lesson 6: Variables Unplugged: Envelope Variables

Unplugged | Variable

Overview

Variables are used as placeholders for values such as numbers or words. Variables allow for a lot of freedom in programming. Instead of having to type out a phrase many times or remember an obscure number, computer scientists can use variables to reference them. This lesson helps to explain what variables are and how we can use them in many different ways. The idea of variables isn't an easy concept to grasp, so we recommend allowing plenty of time for discussion at the end of the lesson.

Purpose

Variables are very helpful in programming. Students will be introduced to this topic using envelopes to represent variables that have been given names. The value of the variable will be written on a card inside of an envelope. This lesson helps students understand how names can be a placeholder for values in the physical world, so that programming with variables will seem less confusing in the virtual world.

Agenda

Warm Up (10 min)

- **Vocabulary Introduction**

Main Activity (20 min)

- **Envelope Variables - Worksheet**

Wrap Up (10 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (10 min)

- **Envelope Variables - Assessment**

Extended Learning

Objectives

Students will be able to:

- Identify variables and determine their values.
- Define and call variables in the context of real-life activities.
- Create situations which require the use of variables.

Preparation

- [ ] Watch the **Envelope Variables - Teacher Video**.
- [ ] Obtain 6 or more blank envelopes for warm up plus some for the main activity.
- [ ] Print one **Envelope Variables - Worksheet** per student.
- [ ] Print one **Envelope Variables - Assessment** for each student.
- [ ] Provide students with envelopes, paper, pens & pencils.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Envelope Variables** - Teacher Video
- **Envelope Variables** - Worksheet
- **Envelope Variables** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Variable** - A placeholder for a piece of information that can change.

Teaching Guide

Warm Up (10 min)

Vocabulary

This lesson has one important word:

- **Variable** - Say it with me: Vayr-ee-ah-buhl

A placeholder for a piece of information that can change.

Introduction

Call four volunteers to the front of the room and line them up. Let the students know that you are going to write a poem for each of them.

On the board (or under your document camera) write the sentence for your first student (suppose it's Bill):

"My student Bill, standing proud
is a fine example for the crowd"

Encourage the students to clap at your abilities and thank Bill for volunteering. Allow Bill to sit down (or go to the back of the line) as you erase the board, then call the next volunteer (we'll say that she's called Annie).

"My student Annie, standing proud
is a fine example for the crowd"

Again, accepting applause, erase the board and invite the next volunteer.

"My student Jenny, standing proud
is a fine example for the crowd"

As you call the final volunteer, inquire as to whether everyone in the class would like a poem written about each of them. Maybe the everyone in the whole school? Goodness, that's going to take a while! Pose the question to your students:

"How could I do this more quickly?"

Your students will likely pick up on the fact that only one word is changing, and that word is simply a person's name. Help them see the location by circling Jenny's name on the board and writing "firstName" next to it.

"It would take a long time to write a poem for everyone in the school if I couldn't start until I knew who I was writing it about, wouldn't it?"

- How long do you think it would take to make a video game if they couldn't start until they knew your username?
- How expensive would video games be if they had to be created separately for each person?
- How do you think we can get around that?

By this time, it's quite likely that your class will come up with the idea of having a placeholder. With that, they're most of the way into understanding where this lesson goes.

- What would we call that placeholder?
  - We need to call it something that makes sense. We wouldn't want to call it "age" if it was a placeholder for their name, right?

Now, let's add some more volunteers. Give them each a piece of paper to write their name on, and have them tuck it inside individual envelopes labeled firstName.

This time, put the poem on the board with a blank space labeled "firstName" where the student's name will go.

- Have the first student in line (likely the last student from the previous example) pull their name from the envelope and that's what you'll write in the space.
- When you erase the board, only erase the portion with the last student's name in it.

- Call the next student to show their variable.
- Repeat as many times as is entertaining

Now it's time for the main activity.

Main Activity (20 min)

Envelope Variables - Worksheet

Once the students understand how the envelopes relate to the sentences, pass out the activity worksheet and let them prepare some variables of their own.

Directions:

- Divide students into groups of 2-4.
- Have students design (draw) a robot.
- After 10-15 minutes, request that the students fill their envelopes with important details about their robot such as its name, height, and purpose.
- Collect each group's envelopes, then bring them to the front of the room to share.
- Write on the board, "My robot's name is robotName, it is numUnitsTall tall, and it's purpose is purpose."
- Use the envelopes to fill the appropriate variable in the sentence, then ask each group to stand when they hear the sentence that describes their creation.

Wrap Up (10 min)

Flash Chat: What did we learn?

- What did we learn today?
- Can you think of anywhere that you have seen variables before?
- There is at least one variable at the top of most homework hand outs? Can you think of what it could be?
- Why do you think that professionals do not put spaces in variable names?
  - What would happen if there was a variable "eye" a variable "color" and a variable "eye color"?
- Variables can be used to store numbers, too.
  - Suppose I have envelopes labeled num1 and num2, then I write num1+num2?
  - What happens if the "num1" envelope contains the number 4 and "num2" contains the number 5?

Journaling

Having students write about what they learned, why it’s useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a variable?
- Why do you think variables are important in programming?

Assessment (10 min)

Envelope Variables - Assessment

Allow students enough time to finish this assessment. If you are willing to spare more time, go over the answers as a class.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

What's in the box?

- Draw boxes on a piece of paper with simple mathematical operators between them.
  - For instance [] + [] = []
- Have similar size squares with numbers between 1 & 20.
- Ask one student to come create a true equation, using the numbers provided.
- Once the student has finished (and the class verifies the equation) exchange one of the numbers with another one, then remove a second number entirely.
  - Tell the students that there is a hidden number in the empty box that makes that equation true again.
  - What number is in the box?
- Play this game over and over again until you can remove the number from any location and the students can figure out what it is supposed to be.

Name: _____ Date: _____

U

Unplugged

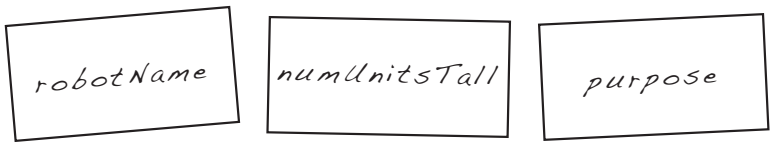
Variables in Envelopes

Robot Variables Worksheet

C O D E

Think about a robot. What is it supposed do? What does it look like?

Draw your robot on paper. When you're done, answer the three questions below on separate pieces of paper, then put them in the correct envelopes.



1. My robot's name is robotName .

2. My robot's hight is numUnitsTall (don't forget units!) .

3. My robot's primary purpose is purpose .

Name: _____ Date: _____

U

Unplugged

Variables in Envelopes

Variables Assessment Worksheet

Given the value of each variable envelope, fill-in the blanks to finish the sentence.

color = pink

petalNumber = 22

animal = monkey

bestSport = golf

hobby = coding

When I grow up, I want to own a guard _____ (animal).

I found a flower with _____ (petalNumber) petals, so I picked it.

My dad just painted his house _____ (color) to match his car.

I love _____ (hobby) . I do it every evening.

There is no such thing as _____ (color) rivers, so if you find one, don't swim in it!

The best sport in the world is _____ (bestSport), do you agree?

Variable envelopes can also contain number values. Use these envelopes and the provided equations to figure out the magic numbers below.

numOne = 2

numTwo = 5

numThree = 7

[ ] (magicNumberA) = _____ (numThree) - _____ (numOne)

[ ] (magicNumberB) = _____ (numTwo) * _____ (numOne)

[ ] (magicNumberC) = _____ (numOne) + _____ (numTwo) * _____ (magicNumberB)

Lesson 9: For Loops Unplugged: For Loop Fun

Unplugged | For Loops

Overview

We know that loops allow us to do things over and over again, but now we're going to learn how to use loops that have extra structures built right in. These new structures will allow students to create code that is more powerful and dynamic.

Purpose

At this point, students have become masters of loops. Today, they will learn about another loop commonly used in programming. The `for` loop repeats commands a certain number of times, but also keeps track of the values it is iterating over. For example, a `for` loop that begins at 4, ends with 8, and has an interval of 1 will repeat 4 times, but the values 4, 5, 6, and 7 will also be captured for use elsewhere. Using this structure with variables can create some pretty fantastic programs. Today, students will simply be learning the basics of a `for` loop before diving into programming with them next time!

Agenda

Warm Up (20 min)

- **Vocabulary**
- **For One and All**

Main Activity (20 min)

- **For Loop Fun - Worksheet**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (5 min)

- **For Loop Fun - Assessment**

Extended Learning

Objectives

Students will be able to:

- Determine starting value, stopping value, and interval of `for loop`.
- Illustrate the counter values hit each time through a `for loop` during runtime.

Preparation

- [ ] Watch the **For Loop Fun - Teacher Video**.
- [ ] Watch the **For Loop Fun - Lesson in Action Video**.
- [ ] Print one **For Loop Fun - Worksheet** per group.
- [ ] Print one **For Loop Fun - Assessment** for each student.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **For Loop Fun** - Teacher Video
- **For Loop Fun** - Lesson in Action Video
- **For Loop Fun** - Worksheet
- **For Loop Fun** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **For Loop** - Loops that have a predetermined beginning, end, and increment (step interval).

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

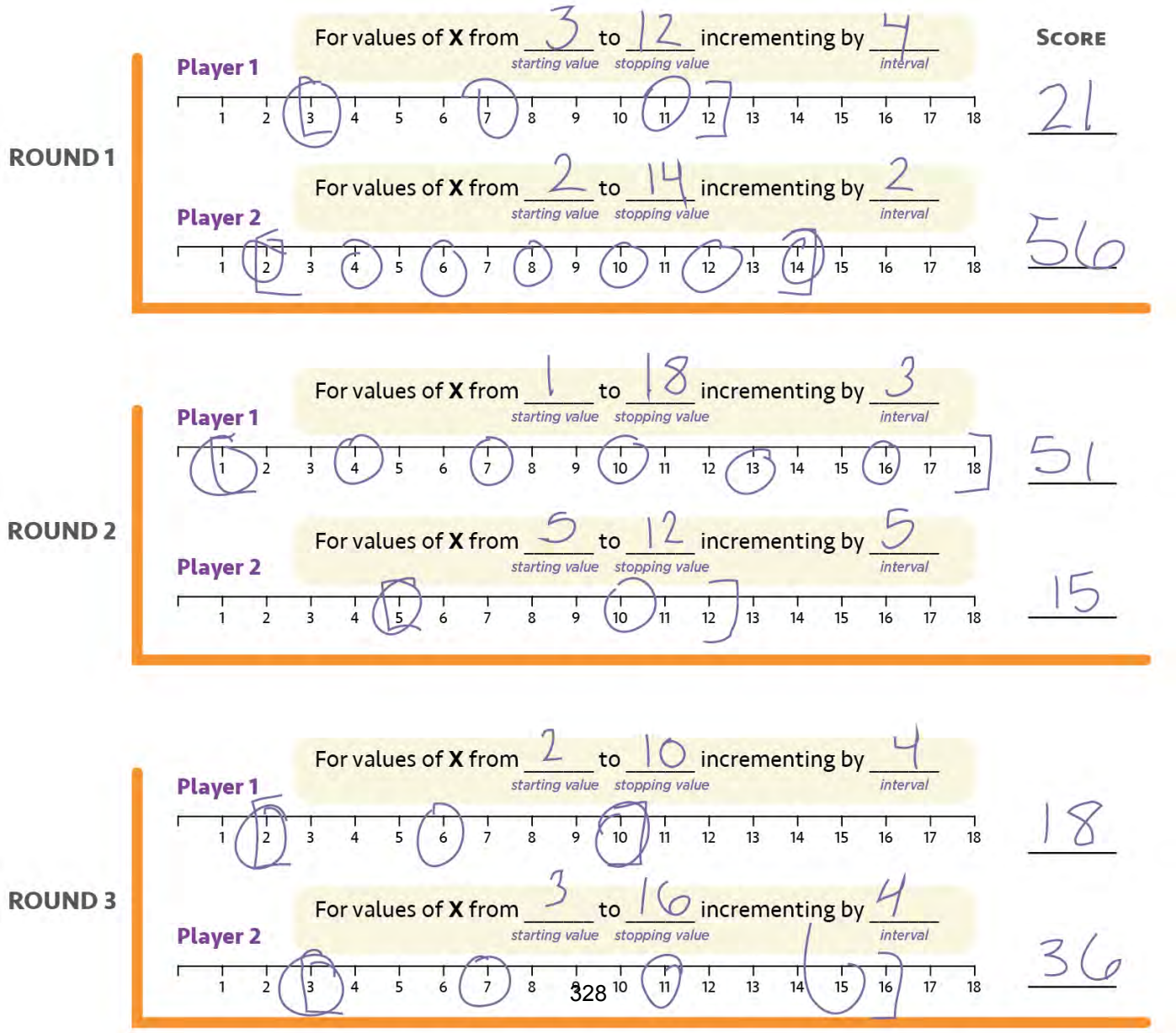
- **For Loop** - Say it with me: For-Loop

Loops that have a predetermined beginning, end, and increment (step interval).

For One and All

- Point out that there are certain loops that happen very frequently, for example, loops where you need to keep track of how many times you have been through
  - Sometimes, you don't want to start with one
  - Sometimes, you don't want to count by ones
  - `for` Loops give you a powerful way to keep a counter that starts when you want, ends when you want, and increases by whatever size step that you want

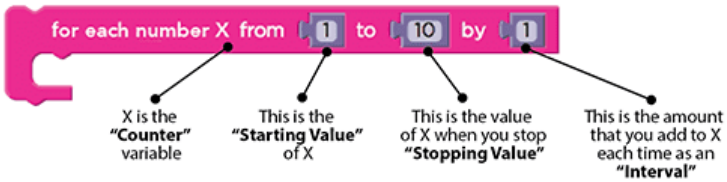
Here, you can jump right into a sample of the game (example in English)



Main Activity (20 min)

For Loop Fun - Worksheet

Sometimes we want to repeat things a certain number of times, but we want to keep track of values as we do. This is where a `for` loop comes in handy. When you use a for loop, you know right from the start what your beginning value is, what your ending value is, and how much the value changes each time through the loop.



`for` Loop block (in English)

Directions:

- Divide students into pairs
- To start the round, each student rolls three times:
  - One die to determine the starting value of X
  - Three dice to determine the stopping value for X
  - One die to determine the step interval of X each time through
- Use one of the provided number lines to trace the for loop that they've made
  - Start at the starting value of X
  - Count down the number line, circling the numbers at the rolled interval
  - Stop when you get to the predetermined stopping value
- Add all of the circled values to your score, then let the other player take a turn
- Best 2 out of 3 wins

Lesson Tip

When you play this game, it's as if you're running through a loop like this

```
for (x=startValue; x <= stopValue; x = x + interval){
circle currentXvalue;
add currentXvalue to roundScore;
}
```

It may be difficult for young students to understand this written in pseudocode, but it may be helpful to have you explain out loud (and perhaps with a diagram) what they will be using as the content of a for loop.

Wrap Up (15 min)

Flash Chat: What did we learn?

- What would your interval need to be if you wanted to count from 4 to 13 by threes?
- What kinds of things do you think you could do with a for loop?
- Can you reproduce a normal loop using a for loop?
- What would you need to do?

Lesson Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow-partner.

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a `for` loop?
- Why would you use a `for` loop instead of a `repeat` loop or a `while` loop?

Assessment (5 min)

For Loop Fun - Assessment

Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Run it Backward

- Try this activity again, but this time have the start number be selected using three dice, and the stop number with only one. Make sure to have a negative increment!

Hop Scotch

- Using chalk, draw a hop scotch diagram outside on the blacktop
  - Number the squares from bottom to top
  - Have students give each other a start square, stop square, and how many at a time they need to jump
  - When the jumper is done, have them write down the loop they just performed
  - Start adding additional activities to be done at each square, this will add complexity to the written portion, as well

Name: _____ Date: _____

U

Unplugged

For Loop Fun

Number Lines and Score Sheet

C O D E

Directions:

* Use the number lines to trace the "for loop" for each turn
    * Start at the starting value of X
    * Count down the number line, circling the numbers at the correct interval
    * Stop when you get to the stopping value
* Add all of the circled values to get the score for your round
* Best 2 out of 3 Wins

ROUND 1

Player 1

For values of **X** from _____ to _____ incrementing by _____
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Player 2

For values of **X** from _____ to _____ incrementing by _____
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Score _____ _____

ROUND 2

Player 1

For values of **X** from _____ to _____ incrementing by _____
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Player 2

For values of **X** from _____ to _____ incrementing by _____
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

_____ _____

ROUND 3

Player 1

For values of **X** from _____ to _____ incrementing by _____
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Player 2

For values of **X** from _____ to _____ incrementing by _____
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

_____ _____

U

Unplugged

Name: _____ Date: _____

For Loop Fun

Sample Game Sheet

C O D E

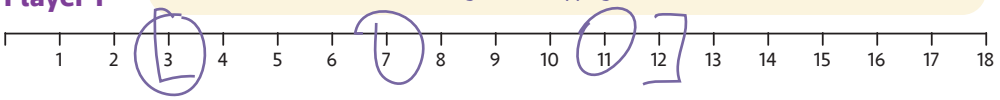
Directions:

* Use the number lines to trace the "for loop" for each turn
  * Start at the starting value of X
  * Count down the number line, circling the numbers at the correct interval
  * Stop when you get to the stopping value
* Add all of the circled values to get the score for your round
* Best 2 out of 3 Wins

ROUND 1

Player 1

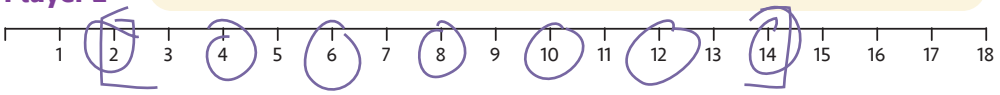
For values of **X** from 3 (starting value) to 12 (stopping value) incrementing by 4 (interval)



Score: 21

Player 2

For values of **X** from 2 (starting value) to 14 (stopping value) incrementing by 2 (interval)

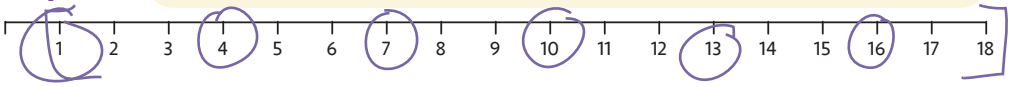


56

ROUND 2

Player 1

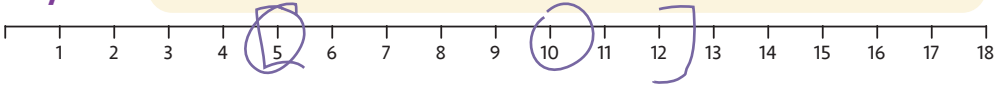
For values of **X** from 1 (starting value) to 18 (stopping value) incrementing by 3 (interval)



51

Player 2

For values of **X** from 5 (starting value) to 12 (stopping value) incrementing by 5 (interval)

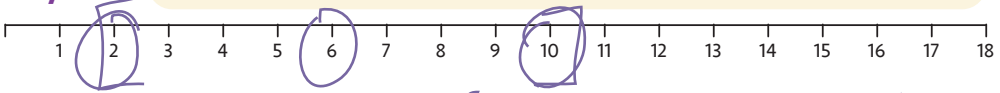


15

ROUND 3

Player 1

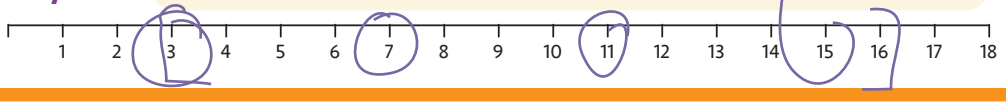
For values of **X** from 2 (starting value) to 10 (stopping value) incrementing by 4 (interval)



18

Player 2

For values of **X** from 3 (starting value) to 16 (stopping value) incrementing by 4 (interval)



36

Unplugged

Name: _____ Date: _____

For Loop Fun

Assessment Worksheet

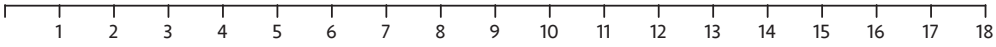
C O D E

Below, you will find three rounds of the For Loop Game, along with what each player rolled during their turn. Fill out the number lines and tally the scores for each round.
Who won the game?

ROUND 1

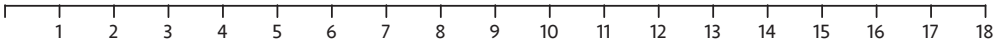
Player 1

For values of **X** from 1 (*starting value*) to 18 (*stopping value*) incrementing by 4 (*interval*)



Player 2

For values of **X** from 3 (*starting value*) to 11 (*stopping value*) incrementing by 2 (*interval*)

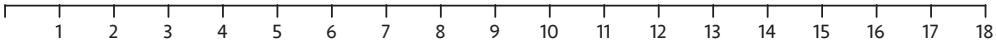


SCORE _____

ROUND 2

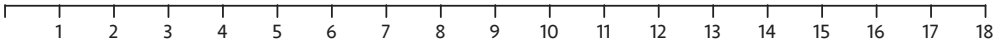
Player 1

For values of **X** from 3 (*starting value*) to 17 (*stopping value*) incrementing by 5 (*interval*)



Player 2

For values of **X** from 5 (*starting value*) to 17 (*stopping value*) incrementing by 3 (*interval*)

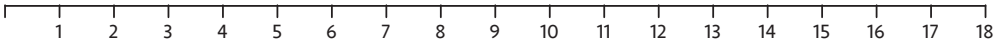


_____

ROUND 3

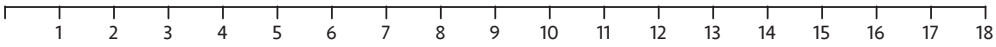
Player 1

For values of **X** from 6 (*starting value*) to 11 (*stopping value*) incrementing by 1 (*interval*)



Player 2

For values of **X** from 2 (*starting value*) to 15 (*stopping value*) incrementing by 6 (*interval*)



_____

WHO WON?
PLAYER # _____

Directions:

* Use the number lines to trace the "for loop" for each turn
    * Start by circling the number at the starting value of X
    * Count down the number line, circling the numbers at the correct interval
    * Stop when you get to the stopping value
* Add all of the circled values to get the score for your round
* Best 2 out of 3 Wins

Lesson 12: Functions Unplugged: Songwriting with Parameters

Unplugged | Function | Parameter

Overview

One of the most magnificent structures in the computer science world is the function. Functions (sometimes called procedures) are mini programs that you can use over and over inside of your bigger program. This lesson will help students intuitively understand why combining chunks of code into functions is such a helpful practice, and how they can use those structures even when chunks of code are slightly different.

Purpose

Using functions helps simplify code and develops the student's ability to organize their program. Parameters will help the students customize their functions so that they can be used for patterns that are similar, though not identical. Students will quickly recognize that writing functions will make their long programs easier to read and easier to debug if something goes wrong.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **Sing a Song**

Main Activity (20 min)

- **Functions Unplugged: Songwriting With Parameters - Worksheet**

Wrap Up (15 min)

- **Flash Chat: What did we learn?**
- **Journaling**

Assessment (5 min)

- **Functions Unplugged: Songwriting With Parameters - Assessment**

Extended Learning

Objectives

Students will be able to:

- Modify functions to accept parameters.
- Describe how functions and parameters can make programs easier to write.

Preparation

- [ ] Watch the **Functions Unplugged: Songwriting With Parameters - Teacher Video**.
- [ ] Watch the **Functions Unplugged: Songwriting With Parameters - Lesson in Action Video**.
- [ ] Print several **Functions Unplugged: Songwriting With Parameters - Worksheet** for each group.
- [ ] Print one **Functions Unplugged: Songwriting With Parameters - Assessment** for each student.
- [ ] Access to the internet, or pre-downloaded songs and lyrics for activity.
- [ ] Make sure every student has a **Think Spot Journal**.

Links

For the Teacher

- **Functions Unplugged: Songwriting With Parameters** - Teacher Video
- **Functions Unplugged: Songwriting With Parameters** - Lesson in Action Video
- **Functions Unplugged: Songwriting With Parameters** - Worksheet
- **Functions Unplugged: Songwriting With Parameters** - Assessment
- **Think Spot Journal** (**PDF** | **DOCX**)

Vocabulary

- **Function** - A piece of code that you can easily call over and over again.
- **Parameter** - An extra piece of information that you pass to the function to customize it for a specific need.

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has two new and important words:

- **Function** - Say it with me: Func-shun

A piece of code that you can call over and over again

- **Parameter** - Say it with me: Pa-ram-eh-ter

An extra piece of information that you pass to the function to customize it for a specific need

Sing a Song

- Let the class know that today is song day!
- We're going to learn a song together.
  - Start with a simple song either written out or projected on the screen
  - Point to the chorus and be sure that the class knows how it goes before you begin on the rest of the song
  - Blast through the song, singing it with them in the beginning, then see what happens when you get to the part where it calls the chorus

Chorus:
Little bunny Foo Foo
Hopping through the forest
Scooping up the field mice
And bopping 'em on the head
Down came the Fairy
And she said
"Little bunny Foo Foo
I don't wanna see you
Scooping up the field mice
And bopping 'em on the head"

Song:
Chorus

I'll give you 3 chances.
Then I'll turn you into a goon!
The next day. . .

Chorus

I'll give you 2 chances.
Then I'll turn you into a goon!
The next day. . .

Chorus

I'll give you 1 chance.
Then I'll turn you into a goon!
The next day. . .

Chorus

"I gave you two chances.
Now I'll turn you into a goon!"
(POOF!)
And the moral of the story is:
Hare today, goon tomorrow!

Teaching Tip

Little Bunny Foo Foo is being used here as an example only. If your students know this song, feel free to use it. Otherwise, choose an appropriate song that they might be more familiar with (either from music class or the radio.)

- It's quite likely that the majority of the class will sing the lyrics for the chorus when you point to that bit.
  - Stop the song once that happens, and explicitly highlight what just happened
    - You defined the chorus
    - You called the chorus
    - They sang the chorus
- Ask the class why they suppose you only wrote the chorus once at the top of the paper instead of writing it over and over in each place where it is supposed to be sung.
  - What are other benefits of only writing the chorus once when you sing it many times?

Now, imagine that this song is a computer program. Defining a title (like "chorus") for a little piece of code that you use over and over again is called creating a function. This is helpful to computer scientists for the some of the same reasons that it is helpful to songwriters.

Lesson Tip

To add more interest, you can look up the lyrics for some popular songs on the Internet. Show the students that the standard for repeating lyrics is to define the chorus at the top and call it from within the body of the song.

- It saves time not having to write all the code over and over in the program
- If you make a mistake, you only have to change it one place
- The program feels less complicated with the repeating pieces defined just once at the top

What about songs where the chorus isn't exactly the same every time? You can still use a chorus, but you have to have a way to let the singer know what special words you will use for each verse.

- These special words are called parameters.
- In programming, parameters are passed as special instructions to functions like this:

```
chorus(parameter1, parameter2)
```

Feel like this is starting to get complicated? Don't worry. We're going to play with songs a little more to try to really understand how this technique is used!

Main Activity (20 min)

Functions Unplugged: Songwriting With Parameters - Worksheet

A fantastic way to compare functions to something we see in our everyday lives is to look at songs. Songs often have certain groups of lyrics that repeat over and over. We call that a chorus.

Directions:

1. Divide into groups of 4, 5, or 6.
2. Give each group several copies of the Songwriting Worksheet
3. Play a short song for the class that contains a clear chorus that does not change from verse to verse.
4. Challenge the class to identify (and write down) the chorus.
5. Compare results from each group. Did everyone get the same thing?
6. Try the activity again, but this time with a song that changes during each repetition of the chorus. Good examples are: Old MacDonald, Baby Bumblebee, or The Hokey Pokey

Discuss with the class:

- Can the students identify a chorus when some words change?
- How might they use the same idea of calling a chorus when the chorus is different from verse to verse?
- These changing words and phrases are called “parameters” and you can pass them into the chorus like this: chorus(cow, moo)
- Play this game over and over until the class has little trouble identifying the choruses.

Lesson Tip

It's most exciting for students to do this lesson with popular music from the radio, but if you're having a hard time finding appropriate songs where the lyrics repeat exactly, here are a few timeless options:

- **5 Little Monkeys**
- **Old MacDonald**
- **Hokey Pokey**
- **BINGO**
- **Baby Bumble Bee**

It is often easier just to have the class listen to (or watch) the song, then vote on what the chorus is by singing it together, rather than writing the whole thing down. If you choose this method, consider having the class do a written chorus for the final song selection to be sure that the visual learners get proper reinforcement.

Wrap Up (15 min)

Flash Chat: What did we learn?

- Would you rather write lyrics over and over again or define a chorus?
- Do you think it's possible to make multiple choruses for the same song?
- Does it make sense to make a new chorus for every time it's needed in a song?

Lesson Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- How do you see functions being helpful in computer science?
- Describe why parameters are helpful when writing the lyrics for a song where the chorus changes slightly.

Assessment (5 min)

Functions Unplugged: Songwriting With Parameters - Assessment

Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Create Your Song

- Start by creating a chorus together, then repeat it between verses of a song that you develop around it.
- Make a change to the chorus, and ponder how much easier it is to change in just one place.
- Change the chorus again, making it much longer than it was originally.
- Add a second chorus and alternate between them in your verses.
- Add parameters to one of your choruses and see how many more options you have.

Songwriting a Program

- What if we acted out songs instead of singing them? All of the sudden, our chorus would be a function of repeated actions, rather than words.
- Use the concepts of the arrows from the Graph Paper Programming lesson and create a program with lots of repeating instructions.
  - Circle those repeating actions so that the class can see where they are.
  - Define a function called "Chorus" above the program.
  - Cross out everywhere the repeating actions appear in the program and write "Chorus" instead.
- Repeat until the class can go through this process fairly undirected.
- Can you figure out how to pass parameters in this exercise?

Group Name: _____ Date: _____

Songwriting Worksheet Example

Using Lyrics to Explain Functions and Procedures

Song Name: Old MacDonald

Chorus:

Old MacDonald had a farm
e-i-e-i-o
And on that farm he had a **P1**
e-i-e-i-o
With a **P2** here and a **P2** there
Here a **P2**, there a **P2**
Everywhere a **P2**, **P2**

Parameter Examples:

Animal Name	Sound	
(P1)	(P2)	(P3)

Song:

Chorus(Cow, Moo)
Chorus(Pig, Oink)
Chorus(Horse, Neeeeigh)
Old MacDonald had a farm
eeeeeeee-iiiiiiiiii
eeeeeeee-iiiiiiiiii
ohhhhhhhhhhhhh!

U

Unplugged

Group Name: _____ Date: _____

Songwriting Worksheet

Using Lyrics to Explain Functions and Procedures

C O D E

Song Name:

Chorus:

Parameter Examples: _____ (P1) _____ (P2) _____ (P3)

Song:

Name: _____ Date: _____

Songwriting

Lesson Assessment

Look at the lyrics for the two songs below.

If it were your job to write this song as a computer program, what chunk of code would you turn into a function so that you could easily use it over and over again?

Circle the segments of each program that repeat most often. Is everything that you circled exactly the same? What parts are different? Those will need to be parameters.

Finish by filling out the Songwriting Worksheet with the song name, chorus, parameters, and a full version of the song that calls the chorus using the parameters that you chose.

Song: Where is Thumbkin?

Where is Thumbkin?
Where is Thumbkin?
Here I am!
Here I am!
How are you today, sir?
Very well, I thank you.
Run away.
Run away.

Where is Pointer?
Where is Pointer?
Here I am!
Here I am!
How are you today, sir?
Very well, I thank you.
Run away.
Run away.

Where is Middleman?
Where is Middleman?
Here I am!
Here I am!
How are you today, sir?
Very well, I thank you.
Run away.
Run away.

Where is Ringman?
Where is Ringman?
Here I am!
Here I am!
How are you today, sir?
Very well, I thank you.
Run away.
Run away.

Where is Pinkie?
Where is Pinkie?
Here I am!
Here I am!
How are you today, sir?
Very well, I thank you.
Run away.
Run away.

U

Unplugged

Name: _____ Date: _____

Songwriting Worksheet

Lesson 8 Assessment - Finding the Function in a Song

C O D E

Song Name:

Chorus:

Parameter Examples: _____ (P1) _____ (P2) _____ (P3)

Song:

Appendix B: Glossary of Vocabulary

Vocabulary

Encouraging students to learn and use official computer science terms will enable them to communicate correctly and efficiently with others and builds their knowledge such that it can be further developed without having to relearn terms and concepts at a later time. The terms and concepts used in the unplugged lessons are defined using words that young students can understand.

abstraction
Pulling out specific differences to make one solution work for multiple problems.

accessibility
The design of products, devices, services, or environments taking into consideration the ability for all users to access, including people who experience disabilities or those who are limited by older or slower technology.

algorithm
A list of steps to finish a task. A set of instructions that can be performed with or without a computer. For example, the collection of steps to make a peanut butter and jelly sandwich is an algorithm.

binary
A way of representing information using only two options.

binary alphabet
The two options used in your binary code.

bit
A contraction of "Binary Digit". A bit is the single unit of information in a computer, typically represented as a 0 or 1.

block-based programming language
Any programming language that lets users create programs by manipulating “blocks” or graphical programing elements, rather than writing code using text. Examples include Code Studio, Scratch, Blockly, and Swift. (Sometimes called visual coding, drag and drop programming, or graphical programming blocks)

Blockly
The visual programming language used in Code.org's online learning system for K-5 students.

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

byte
the most common fundamental unit of digital data eg. Kilobyte, Megabyte, etc. A single byte is 8 bits-worth of data.

call (a variable)
Use a variable in a program.

call (a function)
This is the piece of code that you add to a program to indicate that the program should run the code inside a function at a certain time.

click
Press the mouse button.

code
One or more commands or algorithm(s) designed to be carried out by a computer. See also: program

command
An instruction for the computer. Many commands put together make up algorithms and computer programs.

computational thinking
Mental processes and strategies that include: decomposition, pattern matching, abstraction, algorithms (decomposing problems into smaller, more manageable problems, finding repeating patterns, abstracting specific differences to make one solution work for multiple problems, and creating step-by-step algorithms).

computer science
Using the power of computers to solve problems.

conditionals
Statements that only run under certain conditions or situations.

crowdsourcing
Getting help from a large group of people to finish something faster.

cyberbullying
Doing something on the internet, usually again and again, to make another personal feel angry, sad, or scared.

data
Information. Often, quantities, characters, or symbols that are the inputs and outputs of computer programs.

debugging
Finding and fixing errors in programs.

decompose
Break a problem down into smaller pieces.

define (a function)
To add code inside a function so that the program knows what it is supposed to do when the function is called.

digital citizen
Someone who acts safely, responsibly, and respectfully online.

digital footprint
The information about someone on the Internet.

DNS (domain name service)
The service that translates URLs to IP addresses.

double-click
Press the mouse button twice very quickly

drag
Click your mouse button and hold as you move the mouse pointer to a new location

drop
Release your mouse button to "let go" of an item that you are dragging

DSL/cable
A method of sending information using telephone or television cables.

event
An action that causes something to happen.

event-handler
A monitor for a specific event or action on a computer. When you write code for an event handler, it will be executed every time that event or action occurs. Many event-handlers respond to human actions such as mouse clicks.

F.A.I.L
First Attempt In Learning

fiber optic cable
A connection that uses light to transmit information.

for loop
A loop with a predetermined beginning, end, and increment (step interval).

frustrated
Feeling annoyed or angry because something is not the way you want it.

function
A piece of code that you can easily call over and over again. Functions are sometimes called ‘procedures.’ A function definition is a segment of code that includes the steps performed in the function. A function call is the code segment, typically within the main logic of the program, which invokes the function.

function call
The piece of code that you add to a program to indicate that the program should run the code inside a function at a certain time.

function definition
The code inside a function that instructs the program on what to do when the function is called.

if-statement
The common programming structure that implements "conditional statements".

input
A way to give information to a computer.

Internet
A group of computers and servers that are connected to each other.

IP address
A number assigned to any item that is connected to the Internet.

iteration
A repetitive action or command typically created with programming loops.

loop
The action of doing something over and over again.

online
Connected to the internet.

output
A way to get information out of a computer.

packets
Small chunks of information that have been carefully formed from larger chunks of information.

pattern matching
Finding similarities between things.

parameter
An extra piece of information that you pass to the function to customize it for a specific need.

persistence
Trying again and again, even when something is very hard.

pixel
Short for "picture element" it is the fundamental unit of a digital image, typically a tiny square or dot which contains a single point of color of a larger image.

program
An algorithm that has been coded into something that can be run by a machine.

programming
The art of creating a program.

repeat
To do something again.

run program
Cause the computer to execute the commands you've written in your program.

search engine
A program that searches for and identifies items in a database that correspond to keywords or characters specified by the user, used especially for finding particular sites on the World Wide Web.

servers
Computers that exist only to provide things to others.

toolbox
The tall grey bar in the middle section of Code.org's online learning system that contains all of the commands you can use to write your program.

trustworthy
Able to be relied on as honest or truthful.

try
Attempt to do something

URL (universal resource locator)
A relatively easy-to-remember address for calling a web page (like www.code.org).

username
A name you make up so that you can see or do things on a website, sometimes called a “screen name.”

variable
A placeholder for a piece of information that can change.

website
A location connected to the Internet that maintains one or more pages on the World Wide Web.

while loop
A loop that continues to repeat while a condition is true.

Wi-Fi
A wireless method of sending information using radio waves.

workspace
The white area on the right side of Code.org's online learning system where you drag and drop commands to build your program.

To view all lesson plans and the online activities associated with these lesson plans, please visit http://studio.code.org.

It is thanks to our generous donors that we were able to develop and can offer this course at no cost to schools, teachers, or students:

Microsoft, Infosys Foundation USA, Facebook, Omidyar Network, Google, Ballmer Family Giving, Ali and Hadi Partovi, Bill and Melinda Gates, BlackRock, Jeff Bezos, John and Ann Doerr, Juniper Networks, Mark Zuckerberg and Priscilla Chan, Quadrivium Foundation, Amazon Web Services, Reid Hoffman, Salesforce, Sean N. Parker Foundation, Smang Family Foundation, Verizon