Instructor Handbook

of

Unplugged and Online Lesson Plans

Code.org Presents:

Computer Science Fundamentals

for

Courses 1 - 4

Table of Contents for Courses 1, 2, 3, and 4

TEACHER RESOURCES

COURSE 1

COURSE 2

COURSE 3

COURSE 4

APPENDIX

Curriculum Overview

Code.org Computer Science Fundamentals

WHO MADE THIS?

We believe that every student should have the opportunity to learn computer science, and the reasons are far more varied than simply having a strong resume. Critical thinking, logic, persistence, and creativity help students excel at problem-solving in all subject areas, no matter what their age.

At Code.org, we recognize that this benefit begins early. That's why we partnered with Thinkersmith and Common Sense Media to develop an exciting and engaging curriculum that allows students to explore the limitless world of technology, beginning in elementary school.

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, 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, Reid Hoffman, Salesforce, Sean N. Parker Foundation, Smang Family Foundation, Verizon

WHO IS THIS FOR?

Kids as young as five years-old!

Computer Science Fundamentals is designed primarily for an elementary school audience, but older students find it to be a helpful starting point as well.

Teachers! (in-school, after-school, or home-school)

This curriculum has been developed for use by all educators of young children. We assume no prior computer science knowledge and have provided clear, detailed lesson plans that can be customized to different situations. Learn with your students!

WHICH COURSE IS FOR ME?

Code.org's Computer Science Fundamentals consists of four courses:

- Course 1: beginners, early-readers (ages 4-6)
- Course 2: beginners, readers (ages 6+)
- Course 3: prerequisite is Course 2 (ages 8+)
- Course 4: prerequisite is Course 3 (ages 10+)

The courses are designed to be flexibly implemented. Rather than a rigid pathway based on grade levels, the courses are based on developmental level and prior experience. Teachers and schools can use the course structure to tailor a course sequence to their students' needs and evolving experience.

COURSE STRUCTURE

Our research-based curriculum uses a spiraling education design, in which concepts and skills are revisited in each course while delving deeper each time.

Each course experience is a blend of online activities and "unplugged" activities (lessons in which students can learn computing concepts with or without a computer). The online experiences are composed of mostly self-guided and self-paced tutorials, which use scaffolded sets of programming instructions to explore and practice algorithmic thinking. The unplugged lessons take a hands-on, often kinesthetic approach, making use of physical manipulatives to model computational concepts.

Each course consists of about 18 to 22 lessons, each lasting between 25 and 45 minutes. They can be taught at a comfortable pace whether in consecutive days as a sub-unit or one day a week for 18 weeks. The content of each course builds conceptually on the previous course, so that a student can progress through all four experiences learning new concepts along the way.

Code.org's K-5 curriculum aligns to CSTA Computer Science Standards and ISTE standards, while some lessons also integrate national Math, English Language Arts, and Science standards.

The use of *Computer Science Themes* scaffolds the development of ideas and recognizes the continual construction of knowledge:

- Algorithms, Loops, Functions, Variables, Conditionals
- Data
- Abstraction, Decomposition, Pattern Recognition
- Computing Practice and Programming (Use of computational tools)
- Computers and Communication Devices (Elements of computing devices and networks)
- Community, Global, and Ethical Impacts

Across the entire K-5 curriculum, students will develop the skills of a computer scientist through the development of *Computational Thinking Practices*:

- Creativity
- Collaboration
- Communication
- Persistence
- Problem Solving

TECHNOLOGY REQUIREMENTS

A computing device and an Internet connection.

We work hard to build an environment that is supported by all modern web browsers on desktops and mobile devices, but you'll have the most success with an up-to-date browser (preferably Firefox, Chrome, or 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.

We've also made all our videos available for download. During the tutorials, there is a green download link for each video, allowing students to download individual videos. If all fails, there is a "Show Notes" tab that provides a storyboard equivalent of the video.

GETTING HELP

The curriculum is completely free for anyone, anywhere to teach. You can get support by visiting support.code.org. If you are a teacher and you'd like to attend a free training on our K-5 Computer Science curriculum, look here for a workshop near you, or look for our online PD at studio.code.org/s/K5-OnlinePD.

Image from "Relay Programming"
Course 2: Stage 9

Throughout this book, you will see dozens of "unplugged" activities that have been created to be run without access to the internet, and often, without the use of any technology at all!

Why Unplugged?

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. Designed for groups or entire classrooms, these computer science activities increase exposure to collaboration, creativity, and solution design (planning).

At Code.org, we use unplugged lessons to introduce each idea group before we present them online. This gives students a shared physical experience for teachers to relate back to when working on complex puzzles.

Can I Skip Unplugged Activities?

We highly suggest that you don't! Unplugged activities might take a little more prep the first time you teach them, but the effects are indescribable. Students react well to physical learning, and a single thirty minute unplugged activity can save many hours of confusion in the computer lab.

How Can I Get Help Understanding Unplugged?

We have gathered some videos and resources for you at code.org/curriculum/unplugged. Visit that site to see helpful videos, or to witness select activities in action!

Skills and Strategies

Code.org K-5 Computer Science

TEACHING COMPUTER SCIENCE

As an elementary teacher, computer science may be a new and foreign subject area. Just like any subject area, there are some tips and tricks unique to computer science in general, and our curriculum specifically, that we think will make your life a bit easier.

In addition to the detailed lesson plans, we've put together some helpful printables for both teachers and students. These guides and handouts can serve as helpful resources each time your students sit down to tackle a new set of puzzles. One the following pages you'll see a flag to let you know which are teacher facing documents and which are for students.

G **Teacher Guide**

H **Student Handout**

Puzzle Solving Recipe

Use the puzzle solving recipe with your students to help them perservere through difficult challenges. While many students will come up with their own approach for tackling the puzzles, following this recipe step-by-step provides structure and direction for students who need it.

Debugging Recipe

Building on the Puzzle Solving Recipe, the Debugging Recipe is a more targeted approach for those times when a student just can't figure it out. The Debugging Recipe can also be a good tool for general use in the classroom when something is "buggy." Has the "lining up for recess" program gone awry? Time to break out the debugging recipe!

Rethinking CS Strategies

This one's strictly for the teachers. We've come up with a list of classroom strategies that can help you deal with differentiation, attention, and collaboration.

Code.org Puzzle Solving Recipe

(Based on Polya's Four Step Problem Solving Process)

These tips will help you get unstuck when solving Code.org puzzles!

Step 1: Understand the Puzzle

- What does the puzzle want you to do?
- Can you talk about the problem in your own words?
- Were you given any code to start?
  - What does it do?
  - Why do you think it's there?
- What is the goal of the puzzle?
- Have you solved any other puzzles that are like this one?

Step 2: Create a Plan *(Pick one or more)*

- Write an algorithm.
- Guess and check as you go.
- Draw a picture of what you want to do.
- Try working backward.
- Solve one small piece at a time.
- Compare to a puzzle that you've already solved.

Step 3: Perform and Perfect the Plan

- Did you solve the puzzle?
- If not, hunt for one error at a time.
- Retest your plan after every change.
- If you start to get frustrated, take a deep breath, or leave your screen for a minute. When you come back, you may see what was causing the trouble!
- Ask questions. Maybe one of your friends can help you figure out where your plan goes awry.

Step 4: Check Your Work

- Does your answer solve the puzzle?
- Did you hit all of the goals of this puzzle?
- Now that you have one way to solve the puzzle, is there an easier way to do it?
- If you change this solution a little, will it work for any other puzzles?
- Could you explain your solution to help

Code.org Puzzle Solving Recipe

(Based on Polya's Four Step Problem Solving Process)

Talk about these steps with your students so they have tools to get themselves unstuck while working on puzzles. Also, use these steps to ask leading questions to students who ask for help.

Step 1: Understand the Puzzle

* Do you understand the situation or puzzle prompt?
* Can you restate the problem in your own words?
* Do you understand the code you are given and why?
    - What role does the code play?
* Do you know what the goal of the puzzle is?
* Is this problem similar to another puzzle you have solved?

Step 2: Create a Plan

* Can one (or more) of the following strategies be used?
    - Guess and test
    - Draw a map
    - Draw a picture
    - Look for a pattern
    - Compare to a previously solved puzzle
    - Solve a simpler problem
* Draw a diagram
* Solve an equivalent problem
* Identify subgoals
* Work backwards

Code.org Puzzle Solving Recipe

(Continued)

Step 3: Perform and Perfect the Plan

* Attempt to solve the puzzle based on the plan that you devised. If the strategy did not succeed, look carefully at the feedback provided by any errors that were created, and modify your plan.

* Test and change your strategy often. Don't be afraid to try solutions before you know that they are perfect. You can often reach the right answer by using trial and error each step of the way.

* Take a walk. If you start to get frustrated, leave the screen. Look away. Think of something else for a little while. When you get back, the answer just might come to you!

* Talk with others. They may be able to give you hints, or even explain how something works so that you can discover a solution.

Step 4: Check Your Work

* Is your solution correct? Does your answer satisfy all goals (number of blocks, use of a particular block/concept)?

* Can you see an easier or more efficient solution?

* Can you see how you can extend your solution to a more general puzzle pattern?

Code.org Debugging Recipe

These debugging tips will help you keep moving when you get stuck!

Work to Avoid Mistakes

- Read the directions.
- What is the goal of the puzzle?
- Take it slow and go one step at a time.
- Can you talk about the problem in your own words?
- Were you given any code to start?
  - What does it do?
  - Why do you think it's there?

Debugging

- Look for problems each step of the way.
- Describe what was supposed to happen.
- Describe what is going wrong.
- Does the difference between what was supposed to happen and what did happen give you any clues?

- Fix one thing at a time, then describe how the result changed.
- Try leaving "breadcrumbs" in your program. You can put clues inside your code (like having your program "say" something) to let you know when each chunk runs.
- Try doing each task as its own chunk, then put all of the pieces together at the end so it is easier to see what each thing does.
- Try at least three ways of fixing problems before you ask for help.
- Talk to a friend. Maybe one of your classmates can help you figure out where your plan goes awry.

Code.org Debugging Recipe

Talk about these steps with your students so they are thoughtful when preparing and troubleshooting their solutions.

Work to Avoid Errors

Step 1: Understand the Task at Hand

* Do you understand the situation or puzzle prompt?
* Can you restate the problem in your own words?
* Do you understand the code you are given and why?
  - What role does the code play?
* Do you know what the goal of the puzzle is?
* Is this problem similar to another puzzle you have solved?

Step 2: Pay Attention to Directions & Work

* Make sure that you review the directions several times while you work. It is possible that you will see something new once you understand the task a little more.

* Check your work frequently to make sure that it behaves the way that you intended.

Step 3: Take it Slow

* When you rush through a task, you're more likely to make mistakes that you could have avoided if you were paying attention.

Step 4: One Step at a Time

* Add only one element at a time, and make sure the solution still works. It's much harder to find mistakes when you add lots of new things all at once.

Code.org Debugging Recipe

(Continued)

Debugging

Step 1: Something's Wrong!

* Watch your progress carefully so that you notice errors as soon as they happen.

Step 2: What Was Supposed to Happen?

* If you noticed that something was wrong, then your solution probably does something different from what was asked.
    - What does it do?
    - What does that tell you?

Step 3: Where Does it First Stop Working?

* Step through little by little until you find the first place that your solution goes wrong, then fix that error.

* Step through again to find the next place it goes wrong, then fix that.

* Repeat until your program works!

Step 4: Hidden Bugs

* If you still can't find your error, try leaving "breadcrumbs" in your program. You can put statements in special places to see which ones activate and which ones don't. Those should provide much needed additional information.

* If you're still stuck, try going away for a bit, then coming back. A fresh point of view can work wonders!

* Sometimes it helps to have an extra set of eyes. If you're out of ideas, see if someone else will come take a look with you.

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.

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.

Course 1

OVERVIEW

Students create computer programs with loops and events and write algorithms for everyday tasks. Through this they learn 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 that they can share. Students starting in Course 1 will be early-readers in the lower elementary grades.

Lesson Sequence

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

#	Lesson Name	Description
1	**Happy Maps**	Students create algorithms (sets of instructions) to move a character through a maze using a single command.
2	**Move It, Move It**	Students learn what it's like to instruct their classmates to move through a maze in their classroom.
3	Jigsaw: Learn to drag and drop	Students gain familiarity with a computer by solving jigsaw puzzles, which accustom them to the Code.org system and also to the idea of dragging and dropping. Students learn how to collaborate with others on assignments at the computer.
4	Maze: Sequence	Students write programs (algorithms for the computer) that get a character through a maze. They'll understand the importance of sequence in the programs they write.
5	Maze: Debugging	Using the same environment as the prior lesson, students are presented with a maze and a pre-written program that fails to get the character to the goal. Students will have to "debug" or fix the pre-written program.
6	**Real-Life Algorithms**	Over the first 5 lessons in this curriculum, students have been writing algorithms. This lesson calls out ways we use algorithms in our daily lives. This lesson also focuses on the bigger picture of computer science and how algorithms play an essential part.
7	Bee: Sequence	Students write programs that move a cartoon bee around that gathers nectar and makes honey. This is a more complex version of Maze.

#	Lesson Name	Description
8	Artist: Sequence	Students write programs that move a character around, drawing a line behind it wherever it goes.
9	**Building a Foundation**	Students build a marshmallow structure using only provided supplies. Structures must complete a task (reach a certain height or bear a certain weight), and students discuss the idea of persisting during a task.
10	Artist: Shapes	Students write programs that draw simple shapes, while describing their position relative to other shapes (above, below, etc).
11	Spelling Bee	Students write programs that moves a Bee around a grid of letters. The path the bee takes spells out simple words.
12	**Getting Loopy**	This lesson introduces the programming concept of loops (repeated instructions) through a dance activity. Students will learn simple choreography and then be instructed to repeat it.
13	Maze: Loops	Students write programs in the Maze that involve using a loop.
14	Bee: Loops	Students write programs in the Bee environment that involve using a loop.
15	**The Big Event**	Students are introduced to the programming concept of “events,” which are actions that a computer constantly monitors for. The teacher will press buttons on a fake remote, and students have to shout specific phases depending on which button is pressed.
16	Play Lab: Create a Story	Students write event-driven programs that create games or tell stories. There are puzzles with certain goals and at the end, students are encouraged to express their creativity to create whatever they’d like.
17	**Going Places Safely**	The Internet is a powerful, but sometimes dangerous place. Teachers introduce to students how to stay safe while navigating the Internet.
18	Artist: Loops	Students write programs that draw interesting and beautiful patterns using loops.

U

UNPLUGGED

Happy Maps

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

At the root of all computer science is something called an algorithm. The word "algorithm" may sound like something complicated, but really it's just a list of instructions that someone can follow to achieve a result. To provide a solid base for the rest of your students' computer science education, we're going to focus on building a secure relationship with algorithms.

TEACHING SUMMARY

Getting Started - 10 minutes

1) Vocabulary
2) Step-by-Step

Activity: Happy Maps - 20 minutes

3) Happy Maps: Single-Step Adventure

Wrap-up - 5 minutes

4) Flash Chat - What did we learn? 5) Vocab Shmocab

Assessment - 10 minutes

6) Move the Flurbs Assessment

LESSON OBJECTIVES

Students will:

- List steps to move character around a map
- Arrange directions to reach predetermined goal
- Predict where character will land, given a list of steps

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Maps and arrows from Happy Maps: Single-Step Adventure
- Game Pieces: Flurbs, Arrows, and Things
- Assessment Worksheet: Move the Flurbs Assessment
- Scissors
- Glue

For the Teacher

- Lesson Video
- Teacher Lesson Guide
- Print one Happy Maps: Single-Step Adventure for each group
- Print one Move the Flurbs Assessment for each student

GETTING STARTED (10 MIN)

1) Vocabulary

This lesson has one new and important word:

Algorithm - Say it with me: Al-go-ri-thm
A list of steps that you can follow to finish a task.

2) Step-by-Step

- 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 one step at a time.
- Explain that many tasks can be described using a specific list of instructions. That list is called an algorithm.
- Challenge your students to work together in small groups to come up with algorithms for their single-step and double-step mazes.

LESSON TIP

Students can work in pairs to create the adventures, then work in pairs to solve the adventures of others. If this feels too chaotic you can work together as a class and create the adventure on a document camera, then work together to solve it.

ACTIVITIES: (20 MIN)

3) Happy Maps: Single-Step Adventure

- This worksheet helps teach students how to think ahead in order to plan a short route from the Flurb's start location to the final location, just one square away.
- Print out an activity packet for every group (ideally 2 to 4 students) and cut the Maps apart. Leave the arrow symbols for the students to cut apart.

- Explain the rules to the class, making sure to emphasise the new word "algorithm."

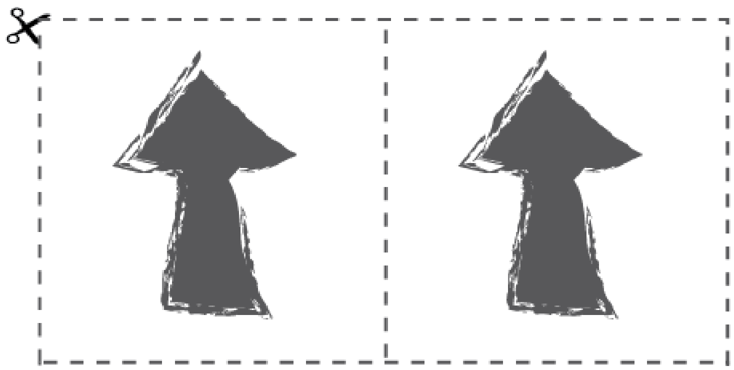
  Flurbs are happy, fuzzy little things.

  Flurbs love to eat fruit. Fruit is hard to find in Flurb Town. Use the maps to help the Flurb find some fruit.

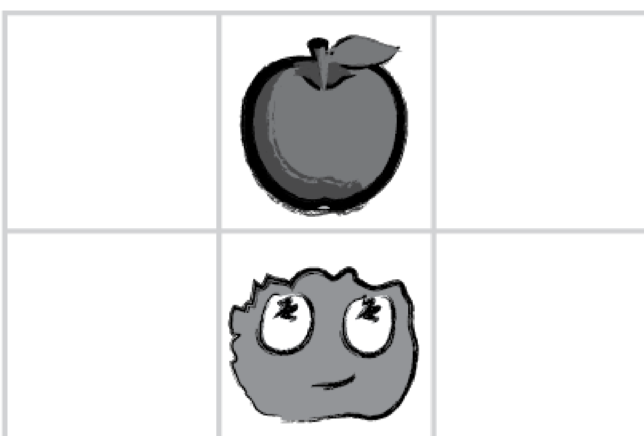
  Work with your group to decide which direction the Flurb needs to step to get to the fruit.

Directions for Class:

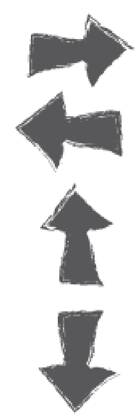
1) Cut out an arrow for each member of your team.



2) Start with Map 1 to help the Flurb look for fruit.



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



3) Have each member of your group put an arrow next to the map to vote for which way the Flurb should step.

4) If not all arrows are pointing the same way, talk to each other and decide as a group which way the arrow should point.

5) When your whole group agrees on a direction, your team can share your answer with the teacher.

6) If your answer is correct, move on to the next map.

WRAP-UP (5 MIN)

4) Flash Chat: What did we learn?

- Did you feel like you were actually telling the Flurb what to do?
- What would it be like to control a robot that way?
- What would you create if it were that easy to tell a computer what 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.

5) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

"Breaking something into exactly two pieces"
"A list of steps that you can follow to finish a task"
"The plastic coating on the end of a shoelace"

...and what is the word that we learned?

ASSESSMENT (10 MIN)

6) Move the Flurbs

- Hand out the worksheet titled "Move the Flurbs" and allow students to complete the activity independently after the instructions have been well explained.

EXTENDED LEARNING

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

Create Your Own

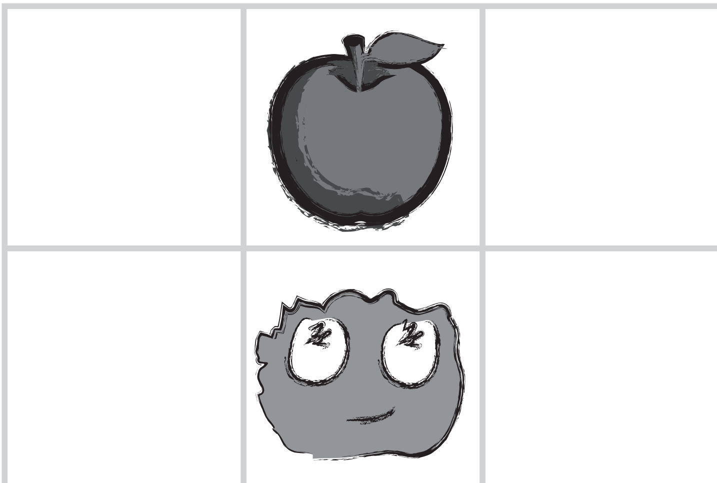
- Allow the students to guide you toward solving a problem (that you provide) one step at a time. Point out that every time they make a step, the rest of the adventure gets easier. If the students are still excited by the exercise, give them a more complicated configuration to solve.

Flurb Flash

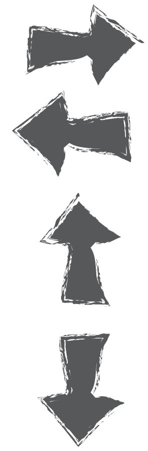
- Cycle quickly through single-step puzzles on your projector. Have the students hold up an arrow card or simply point in the direction that they think the Flurb should move.

1

Happy Map 1

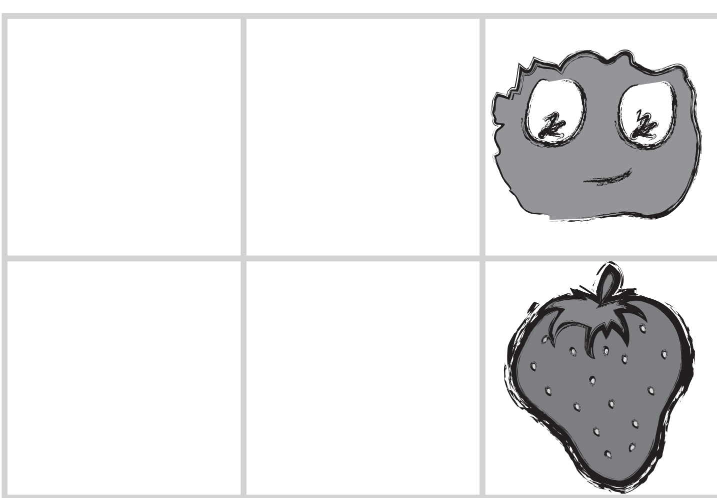


Which way should the Flurb step to get to the fruit?

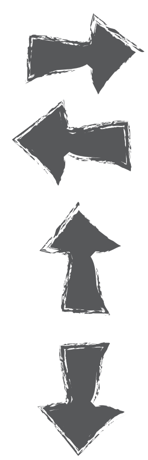


2

Happy Map 2

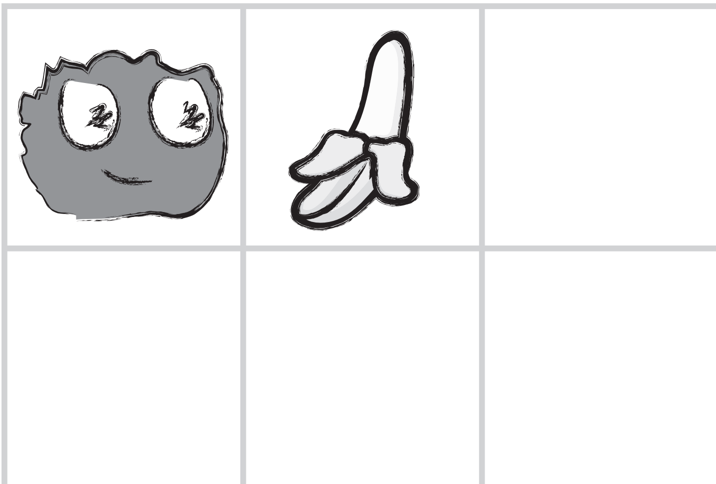


Which way should the Flurb step to get to the fruit?

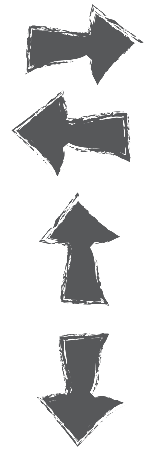


3

Happy Map 3

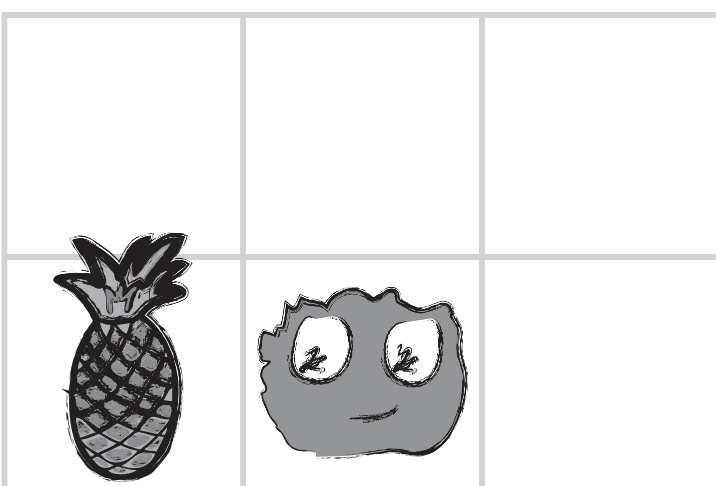


Which way should the Flurb step to get to the fruit?

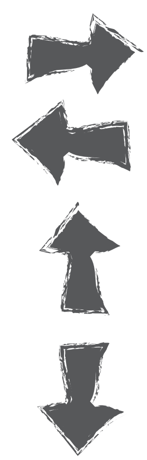


4

Happy Map 4



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



Name: ______________________ Date: ______________

Unplugged

Happy Maps

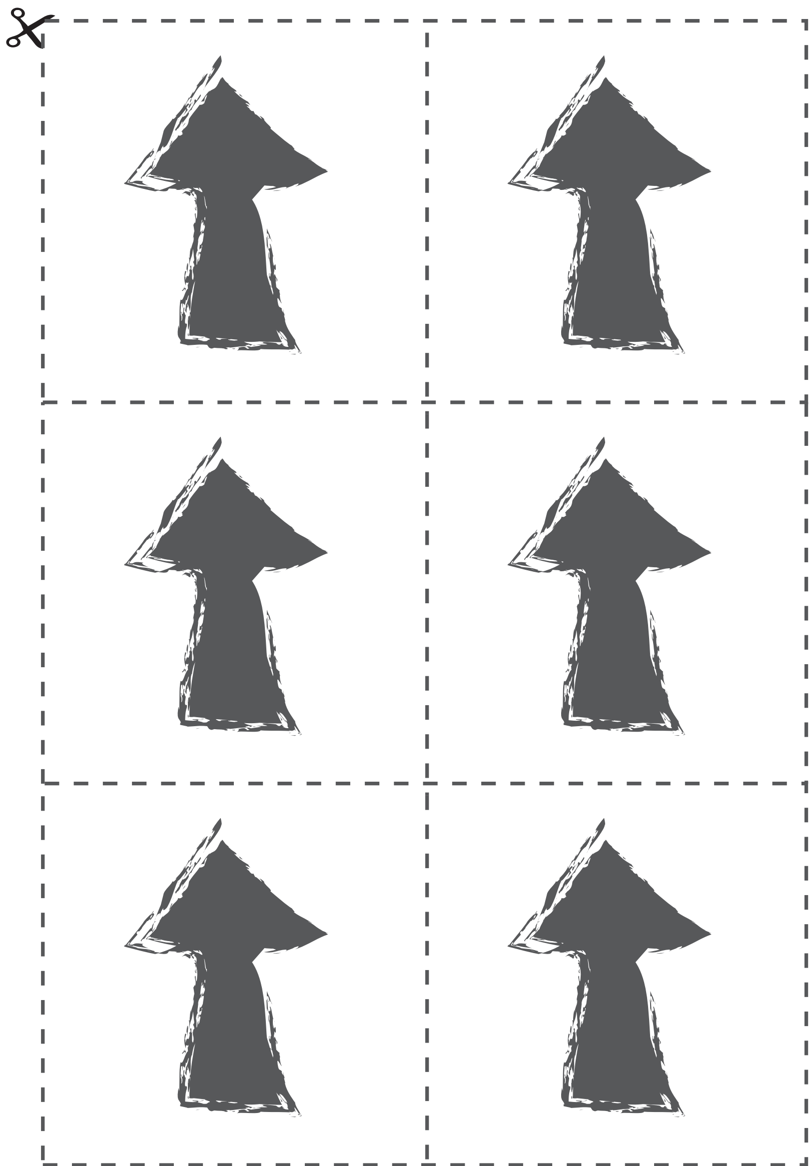
Single-Step Adventure Activity Pieces

Cut out an arrow for each member of your team.

Place your arrow next to the map to vote for which way the Flurb should step.

If not all arrows are pointing the same way, talk to each other and decide as a group which way the arrow should point.

When your whole group agrees on a direction, your team can move on to the next map.



Name: ______________________ Date: ________________

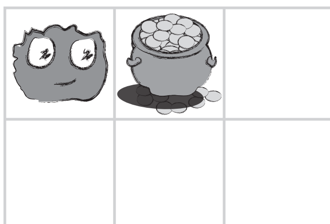
Unplugged

Move the Flurbs

Assessment Worksheet

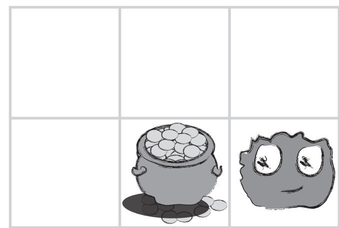
The Flurb's pot of gold is in danger! Help her get to it as quickly as possible before it disappears.

To show the Flurb how to get to her pot of gold, cut out the correct arrows from the bottom of the page and paste them in the program slots by each of the picture maps.



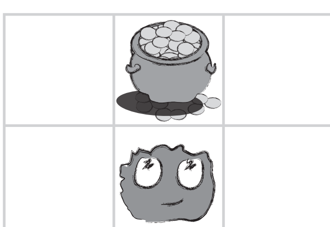
PROGRAM 1

STEP 1



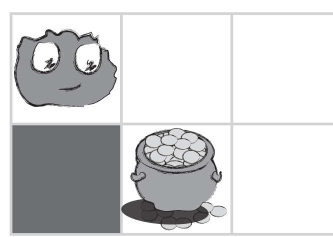
PROGRAM 2

STEP 1



PROGRAM 3

STEP 1



EXTRA CREDIT PROGRAM

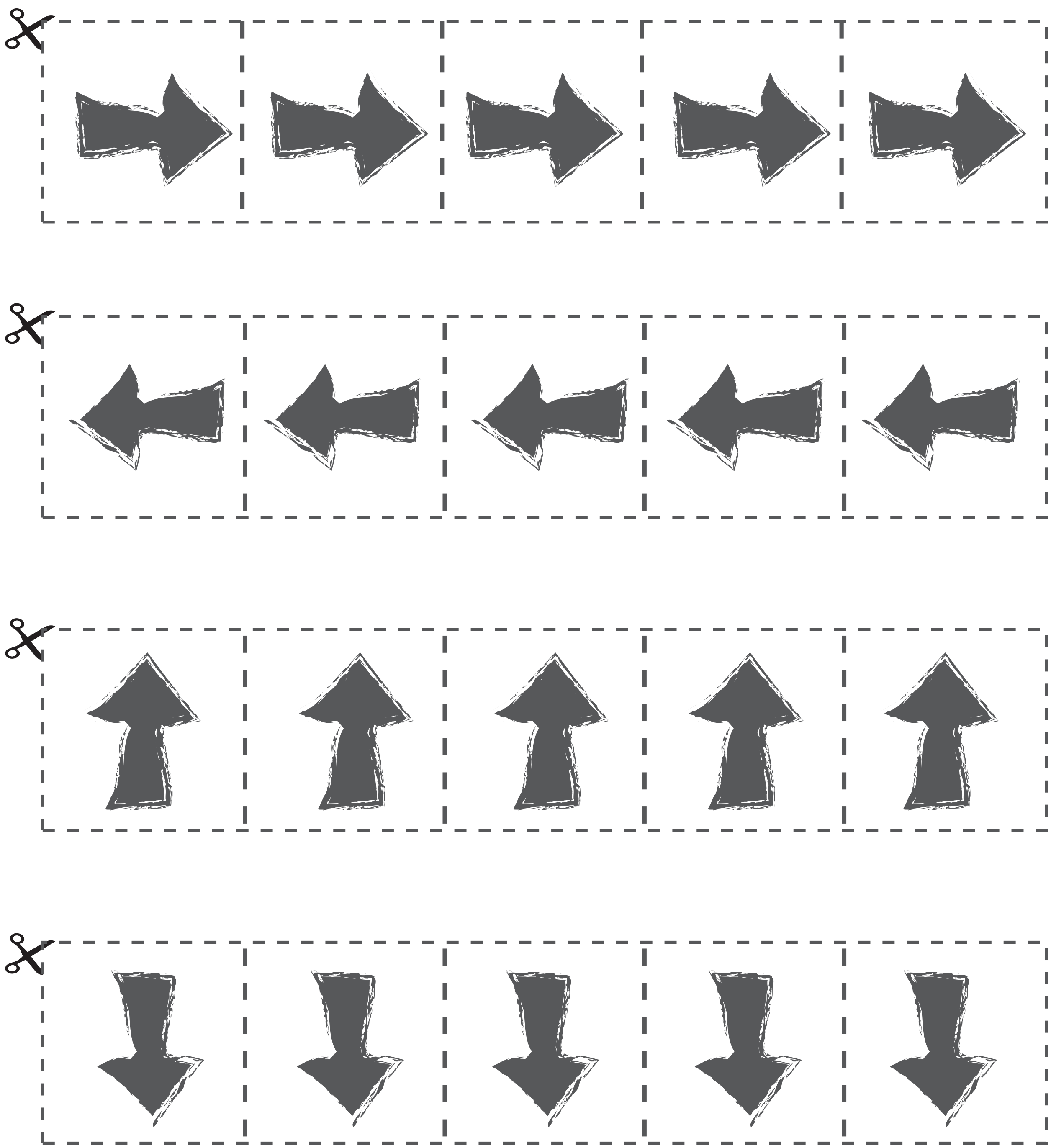
STEP 1	STEP 2

Name: ______________________ Date: _______________

Unplugged

Move the Flurbs

Assessment Worksheet



U

UNPLUGGED

Move It, Move It

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

This lesson will help students realize that in order to give clear instructions, they need a common language. Students will practice controlling one another using a simple combination of hand gestures. Once they understand the language, they will begin to "program" one another by giving multiple instructions in advance.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) Let's Control Ourselves

Activity: Move It, Move It - 20 minutes

4) Move It, Move It: Multi-Step Adventure

Wrap-up - 5 minutes

5) Flash Chat - What did we learn?
6) Vocab Shmocab

Assessment - 10 minutes

7) Move the Flurbs 2

LESSON OBJECTIVES

Students will:

- Recognize situations where they can create programs to complete tasks
- Predict moves necessary to get teammate from start to finish
- Convert movements into symbolic instructions
- Relate algorithms as programs to teammates

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Maps and Key from Move It, Move It: Multi-Step Adventure
- Assessment Worksheet: Move the Flurbs 2
- Scissors

- Glue

For the Teacher

- Lesson Video
- Teacher Lesson Guide
- Print one Move It, Move It: Multi-Step Adventure activity pack on Cardstock for each group
- Print Assessment Worksheet: Move the Flurbs 2 for each student

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. You can do this as one large group or have students discuss with an elbow partner.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) Vocabulary

This lesson has one new and important word:

Program - Say it with me: Pro-gram
An algorithm that has been coded into something that can be run by a machine.

3) Let's Control Ourselves

- Ask your students if they remember helping you draw a smiley face last time.
  - Review the instructions by drawing as they guide you one step at a time.

- Let them know that you're going to make a small change.
    - Now, I want you to give me two instructions at a time before I move my pen.

- Can you do it again, but give me three instructions at a time?

- When you give me multiple instructions at a time, you're providing me with an "algorithm" to draw each piece of the smiley face.

- Now, suppose we were to have a secret "code" for each of those instructions. For example, "Draw an Eye" could look like this: (make a large circle with your hands). If we had special codes for each of those steps, then our algorithm would become a program.

- We're going to play a game that allows us to program each other...and you'll do it all with your arms!

LESSON TIP

Feel free to do an example map with the students as a class before breaking them into groups or even describing the rules (beyond how to react to each of the arm gestures). Learning through play is often more effective than spouting off all of the rules at this age.

ACTIVITY: (20 MIN)

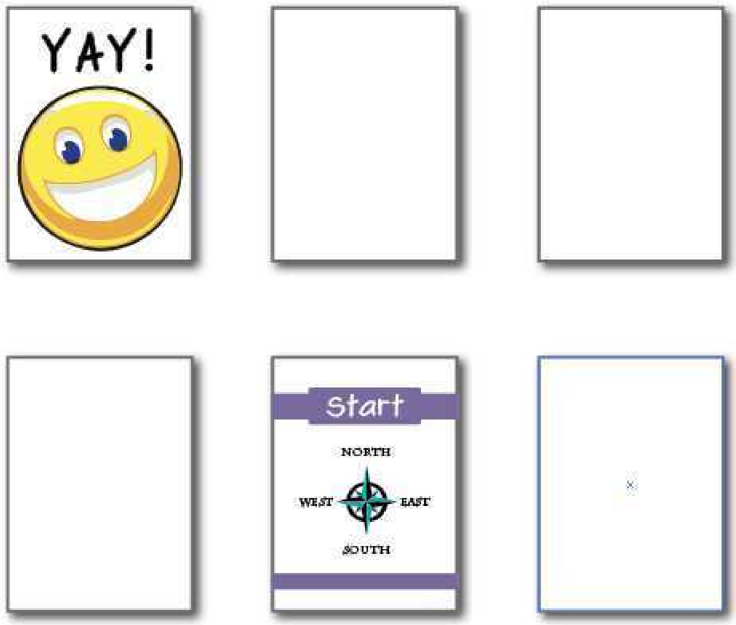
4) Move It, Move It: Multi-Step Adventure

- This worksheet helps teach students how to think ahead in multiple steps, as they plan a short route from their friend's start location to the hidden smiley face, up to three steps away.
- Print out an activity packet for every group (ideally 2 to 4 students) and cut the Map Cards apart.
- Explain the rules to the class, making sure to emphasise the new word "program."

Directions for Class:

1. Decide who will be the Walking Machine and who will be the Controller.
2. Have the Controller set up a grid on the floor made up of pieces of paper as shown on one of the Move It Maps, except with the smiley face upside down, facing the ground.
3. The Walking Machine will start by standing on the page with the compass rose.
4. The Controller will then lead the Walking Machine step-by-step through the paper maze that they created, using the provided arm signals.
5. When the Controller gives the signal to “STOP,” the Walking Machine will flip over the page that they are on. If that page is a smiley face, then the maze was a success!

- The Controller (and anyone else in the group who is not the Walking Machine) can set up a map made of paper, based on one of the Move It Map cards.



- Remember that the smiley face map page should actually be set facing the ground, so that the Walking Machine cannot easily tell where their final location is.
- The Walking Machine begins by standing on the piece of paper imprinted with the compass rose.

LESSON TIP

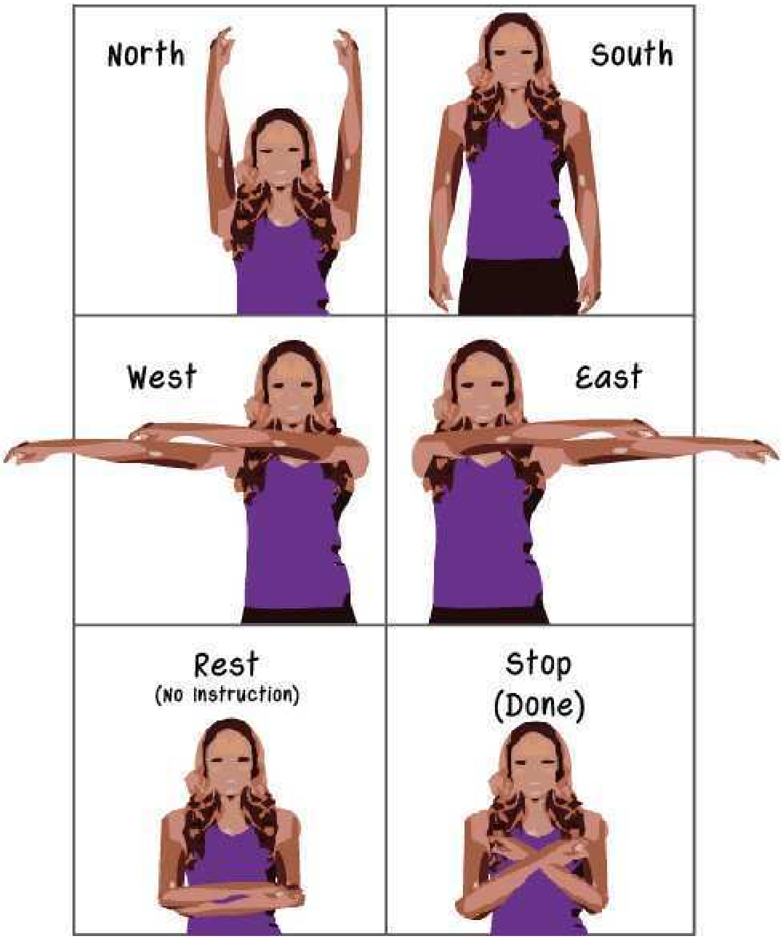
Here are some useful links in case your class hasn't yet talked about the compass rose and cardinal directions:

- The Cardinal Directions Geography Song
- Cardinal Direction Mnemonics Lesson

These topics will be important in the online lessons that follow, so taking an extra couple of minutes to be sure that your students correlate North with Up, South with Down, East with Right, and West with Left will continue to be helpful hereafter.

- The Controller uses arm movements to guide the Walking Machine. Encourage the Controller to be facing the same direction as the Walking Machine to avoid having them get confused by "East is Right" and "West is Left."

- Controllers should start by giving one direction at a time, allowing the Walking Machine to take a step before they move on to the next direction.
    - Halfway into the activity, you can encourage your students to Control with two instructions before they allow the Walking Machine to take a step, and then three.
    - Ideally, by the time the lesson is complete, the students will relay the entire "program" to the Walking Machine before the Walking Machine even takes their first step.



LESSON TIP

Note that the rules are not the most important thing here. Feel free to clarify if the students have questions, but if the students are playing a bit differently than described, you don't need to hold them to the letter of the game. The crucial bit is that they are moving from immediate instructions to giving two or three instructions before the Walking Machine moves.

WRAP-UP (5 MIN)

5) Flash Chat: What did we learn?

- In the game we just played, who do you suppose was more like a programmer, and who was more like a computer?
- What were the four directions on the compass rose?
    - What tricks can we use to remember North, South, East and West?
- How could we have given instructions without using our arms?
- What was your favorite part about that game?

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

  "The fluffy feathers of a baby bird"
  "The circuit board that controls a robot"
  "An algorithm that has been coded into something that can be run by a machine"

  ...and what is the word that we learned?

ASSESSMENT (10 MIN)

7) Move the Flurbs 2

- Hand out the worksheet titled "Move the Flurbs 2" and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous two activities.

EXTENDED LEARNING

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

X's and O's

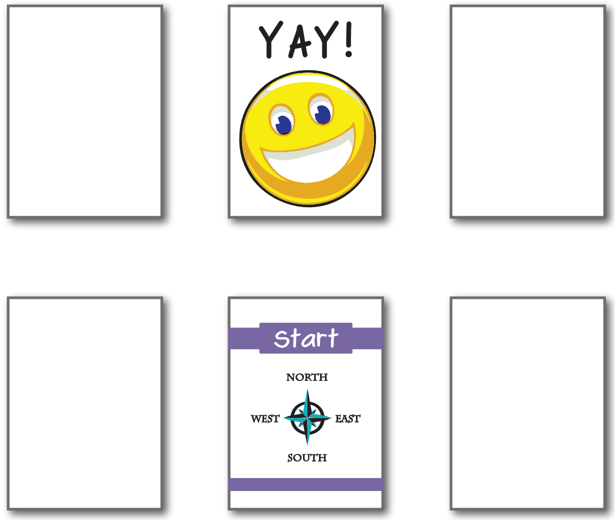
- Draw a tic-tac-toe board for the class.
- Place a single X and a single O somewhere on the board.
- Ask the class if they can get the X to the O using arm gestures as a class.

X's, O's, and Arrows

- Similar to the activity above, but have the students write their programs in advance using arrows instead of hand gestures.
  - This can be done in groups.
  - Groups can share their solutions for the class.

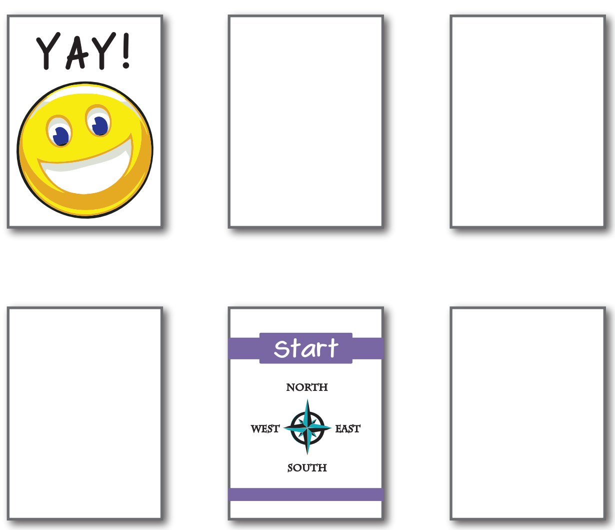
1

Move It Map 1



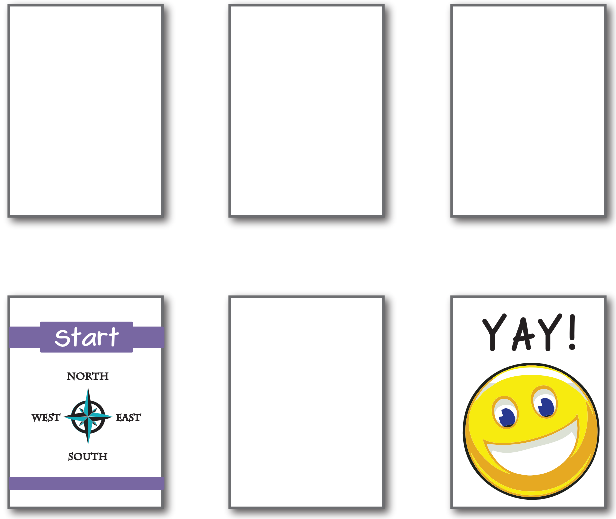
2

Move It Map 2



3

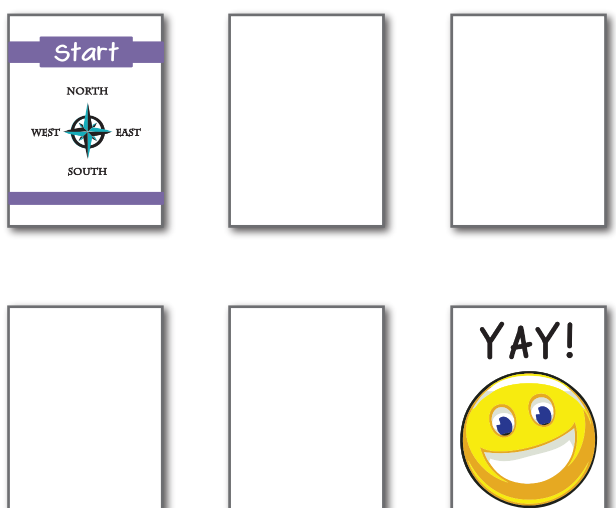
Move It Map 3



Revision 140710.1a

4

Move It Map 4



Revision 140710.1a

Name: ______________________ Date: ______________

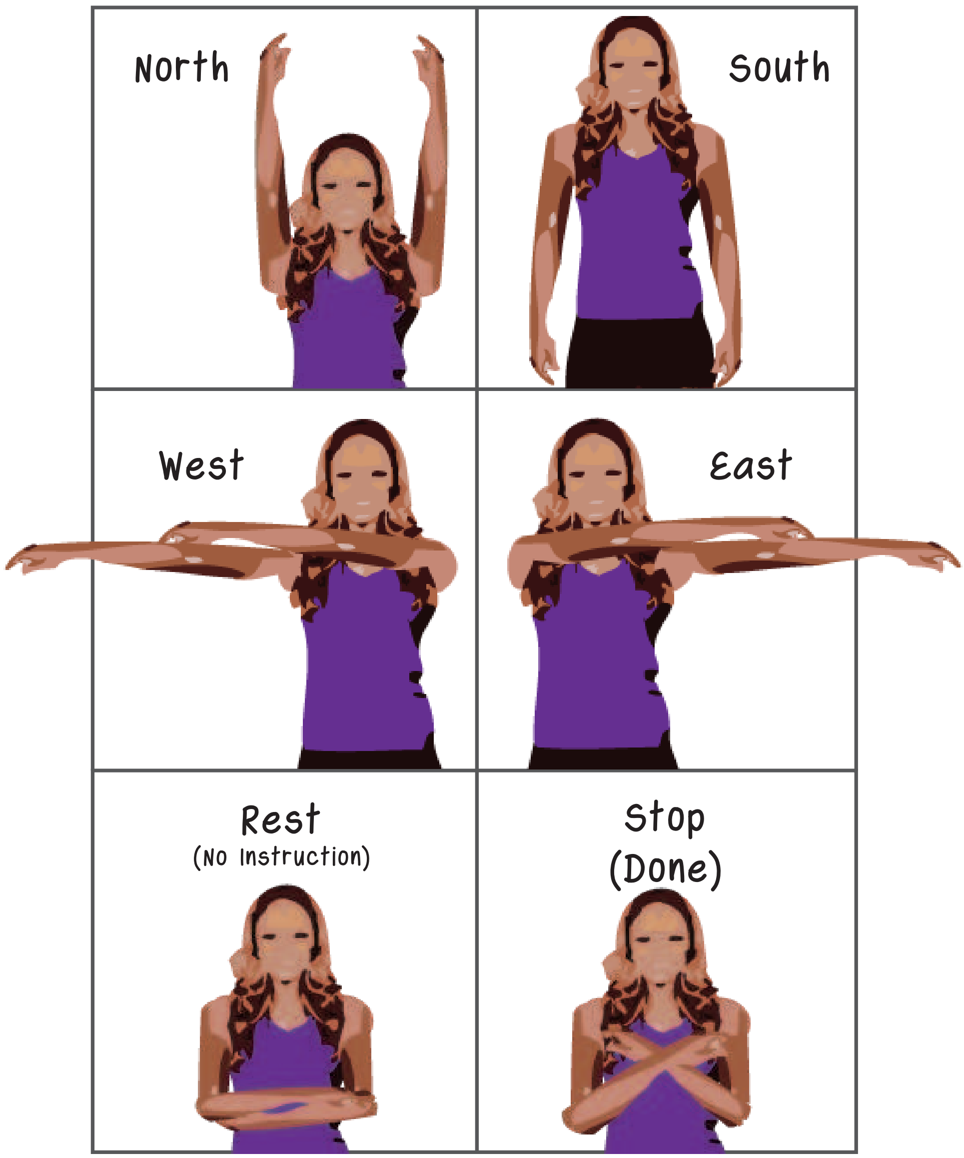
Unplugged

Move It, Move It

Multi-Step Adventure Activity Key

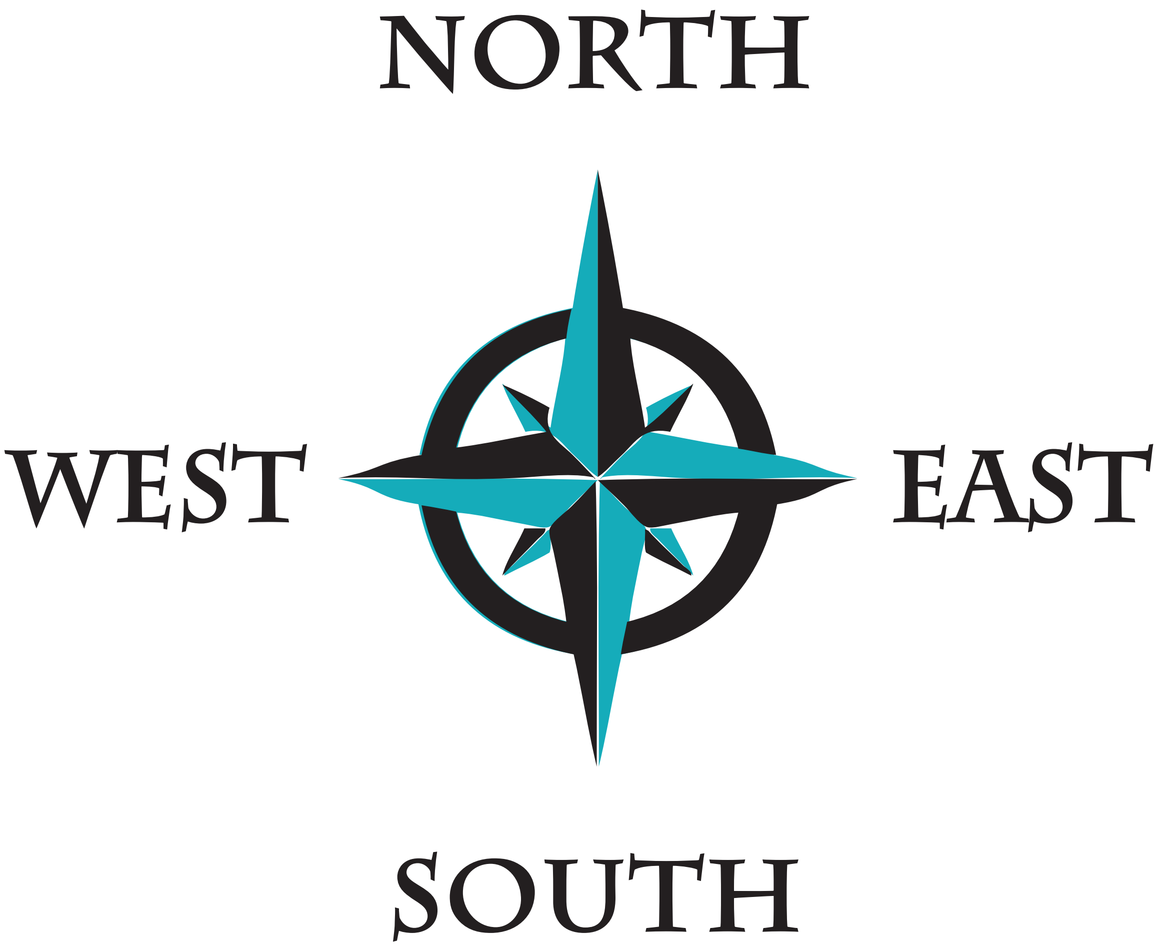
These are the moves that you can do to help guide your friend.

Practice a few times to be sure that you both understand what each move does.



YAY!

Start



Name: ____________________ Date: ______________

U

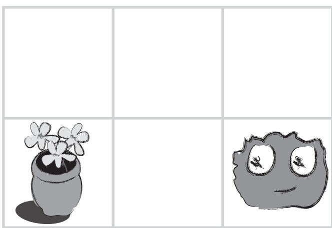
Unplugged

Move the Flurbs 2

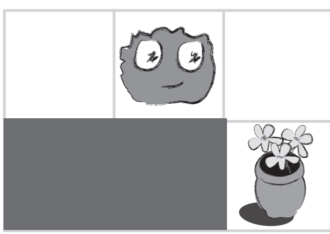
Assessment Worksheet

The weather is getting hot. Help the Flurb get to her flowers so she can water them.

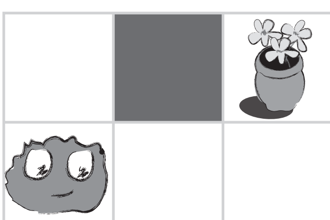
To show the Flurb how to get to her flowers, cut out the correct arrows from the bottom of the page and paste them in the program slots by each of the picture maps.



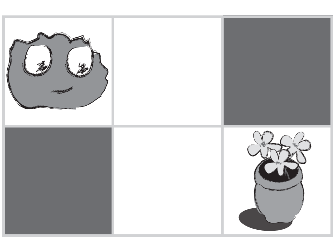
STEP 1	STEP 2



STEP 1	STEP 2



STEP 1	STEP 2	STEP 3



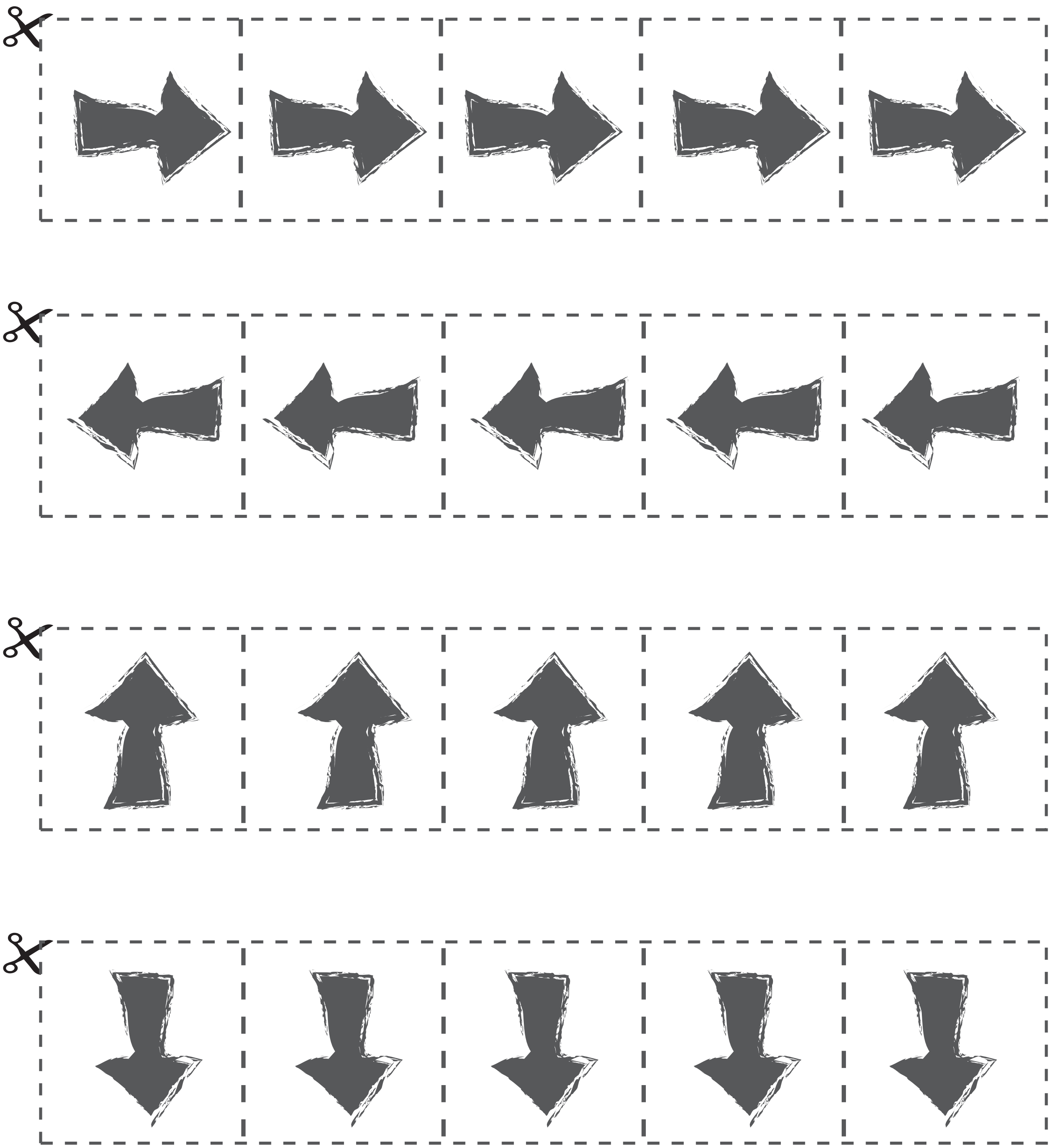
STEP 1	STEP 2	STEP 3

Name: ______________________ Date: ______________

Unplugged

Move the Flurbs 2

Assessment Worksheet



3

Course 1 | Lesson 3

Jigsaw: Learn to drag and drop

Lesson time: 30 Minutes

K-1 LESSON OVERVIEW

In this first online activity students will become familiar with basic mouse use and the block-based programming interface they will be using throughout the course. Students will begin by simply dragging images on screen and then progress to dragging puzzle pieces into the proper order.

LESSON OBJECTIVES

Students will:

- Use a mouse to input information into a computer
- Recall and apply the rules of pair programming
- Use pair programming to complete collaborative tasks with or without a computer
- Identify situations when the rules of pair programming are not followed
- Arrange puzzle pieces into the proper order

GETTING STARTED

Introduction

Ask students if they have ever used a computer mouse before (or trackpad, touchscreen, whatever device they'll be using).

- What does the mouse do?
- How does moving the mouse affect the cursor on the screen?
- You will now be using the mouse to put together puzzle pieces on the computer.
- Pushing down on the mouse button is like closing your hand on a real puzzle piece and letting go of the button is like opening your hand to drop the piece. Make sure you keep your mouse button pressed while you drag each piece.
- Watch the pair programming video together and discuss the role of the driver and navigator.

ACTIVITY

[Jigsaw: Learn to drag and drop]

Pair programming works best with two students to a computer and clear expectations of who is the driver and who is the navigator. You may want to create a visual reminder, like hats or sashes, to identify the driver and the navigator.

LESSON TIP

When puzzle pieces are connected and you want to separate them, you have to drag off from the bottom of the stack. Dragging the top will move the whole stack of pieces.

EXTENDED LEARNING

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

Human Computer

Use real puzzle pieces and pair students into the computer and the user. The user "controls" the computer with a mouse to put the puzzle together.

4

Course 1 | Lesson 4

Maze: Sequence

Lesson time: 30 Minutes

LESSON OVERVIEW

In this series of puzzles, students will build on the understanding of algorithms learned in the Happy Maps and Move It Unplugged activities. Featuring characters from the game Angry Birds, students will develop sequential algorithms to move a bird from one side of the maze to the pig at the other side.

LESSON OBJECTIVES

Students will:

- Order movement commands as sequential steps in a program
- Represent an algorithm as a computer program
- Count the number of times an action should be executed and represent it as instructions in a program

GETTING STARTED

Introduction

Ask your students if they are familiar with the game Angry Birds. Explain that they will be writing programs to help an Angry Bird locate a Pig.

- Review cardinal directions.
- Use the NEWS mnemonic and let students know that they will see those letters in their programs next to the direction arrows.
- Getting the bird to the pig will require putting your directions in a very specific order or sequence.
- Can you solve the puzzles using the fewest blocks possible?

LESSON TIP *Encourage students to continue using the pair programming method that they learned in the last lesson. Ask them to restate the roles of the driver and navigator.*

ACTIVITY

Maze: Sequence

As your students work through the puzzles, observe how they plan the path for the bird. Identify different strategies used and ask students to share with the whole class. This helps students to recognize that there are many ways to approach these problems. You may want to go through a few puzzles on the projector. While doing this you can ask a one student to trace the path on the screen while another writes the directions on a whiteboard.

EXTENDED LEARNING

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

Create Your Own

In small groups, let students design their own mazes and challenge each other to write programs to solve them. For added fun, make life-size mazes with students as the pig and bird.

5

Course 1 | Lesson 5

Maze: Debugging

Lesson time: 30 Minutes

LESSON OVERVIEW

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 missing blocks, extra blocks, and misordered blocks.

LESSON OBJECTIVES

Students will:

- Predict where a program will fail
- Modify an existing program to solve errors
- Identify an algorithm that is unsuccessful when the steps are out of order

GETTING STARTED

Introduction

Ask students about problems they solve in everyday life.

- How do you fix something that isn't working?
- Do you follow a specific series of steps?
- The puzzles in this unit have already been solved for you (yay!), but they don't seem to be working (boo!)
- We call the problems in these programs "bugs," and it will be your job to "debug" them.

ACTIVITY

Maze: Debugging

As your students work through the puzzles, observe how they search for bugs. Identify different strategies used and ask students to share with the whole class. This helps students to recognize that there are many ways to approach these problems.

EXTENDED LEARNING

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

Planting bugs

Have students go back through previous levels, purposefully adding bugs to their solutions. They can then ask other students to debug their work. This can also be done with paper puzzles.

U

UNPLUGGED

Real-Life Algorithms: Plant a Seed

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON 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.

LESSON OBJECTIVES

Students will:

- Name various activities that make up their day
- Decompose large activities into a series of smaller events
- Arrange sequential events into their logical order

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Components for Planting Seeds: Container (such as empty milk carton), potting soil, seed, water
- Real-Life Algorithms: Plant a Seed Worksheet

- Assessment Worksheet: Daily Algorithms
- Scissors
- Glue

For the Teacher

- Lesson Video
- Teacher Lesson Guide
- Print one Real-Life Algorithms: Plant a Seed Worksheet for each student
- Print Assessment Worksheet: Daily Algorithms for each student

GETTING STARTED (10 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. You can do this as one large group or have students discuss with an elbow partner.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) 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

3) 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!

ACTIVITY: (20 MIN)

4) Real-Life Algorithms: Plant a 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.*

- 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:

  1. Cut out the steps for planting a seed from the provided worksheet.
  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 plant their seed!

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.*

WRAP-UP (5 MIN)

5) 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?

ASSESSMENT (15 MIN)

6) Assessment Worksheet: Daily Algorithms

- Hand out the worksheet titled "Daily Algorithms" 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.

U

Unplugged

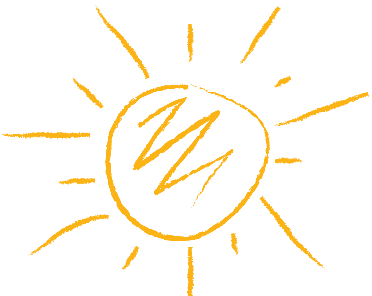
Name: ______________________ Date: ______________

Real-Life Algorithms

Plant a Seed Worksheet

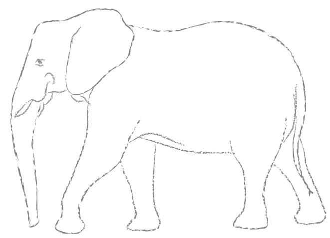
You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to help each other plant a seed.

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!

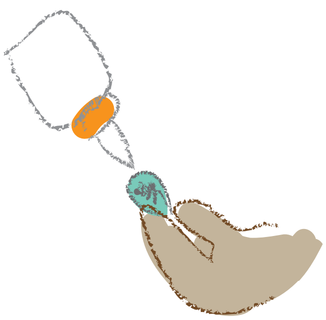


PUT POT IN SUNLIGHT

PUT SEED IN HOLE



HUG AN ELEPHANT



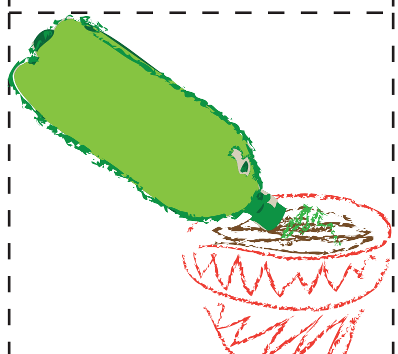
PUT GLUE ON SEED

FILL POT WITH SOIL

POKE HOLE IN SOIL

WATER POT

COVER SEED WITH SOIL



POUR SODA POP IN POT

7

Course 1 | Lesson 7

Bee: Sequence

Lesson time: 30 Minutes

LESSON OVERVIEW

In this lesson students will help their bees to collect nectar from flowers and create honey in honeycombs. This builds on the Maze levels by adding action blocks to the movement blocks students are already familiar with.

LESSON OBJECTIVES

Students will:

- Express movement as a series of commands.
- Order movement commands as sequential steps in a program.
- Represent an algorithm as a computer program.
- Convert a whole number to the equivalent quantity of individual blocks.
- Distinguish between flowers and honeycombs.
- Express the relationships between flowers, nectar, honeycombs, and honey.

GETTING STARTED

Introduction

- Poll students for prior knowledge about bees.
  - Bees collect nectar from flowers and create honey in honeycombs.
  - They communicate using intricate dance steps, which is similar to the instruction steps they'll be using to control their bee.
- In addition to moving, you'll also be using some new code to let your bee collect nectar and make honey.
  - When you see a flower, collect nectar.
  - When you see a honeycomb, make honey.

ACTIVITY

Bee: Sequence

Point out to students that each flower and honeycomb has a little number next to it. That number tells you how much nectar to collect or honey to make.

- Have students count their nectar and honey blocks out loud.

8

Course 1 | Lesson 8

Artist: Sequence

Lesson time: 30 Minutes

LESSON OVERVIEW

In this lesson students will take control of the Artist to complete simple drawings on the screen.

LESSON OBJECTIVES

Students will:

- Create a program to complete an image using sequential steps
- Select an argument for a given command
- Choose the appropriate blocks to draw images with non-continuous lines

GETTING STARTED

Introduction

Brainstorm with students ways to tell someone else how to draw a picture:

- How would you do that with a computer?
- In these puzzles you will be moving a character who leaves a line everywhere it goes.
  - You'll use the cardinal directions to do this, just like we've been doing to move the bird and bee.

ACTIVITY

Artist: Sequence

If students struggle to use the correct number of blocks to draw a line, point out that each line segment has a dot on both ends.

EXTENDED LEARNING

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

The Copy Machine

- Give students two pieces of paper.
- On one sheet, draw a simple image (right angles and straight lines only).
- On the second sheet, draw instructions for recreating that image using a series of arrows.
- Trade instruction sheets and attempt to recreate the image using only the provided instructions.

U

UNPLUGGED

Building a Foundation

Lesson time: 30 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON 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.

This lesson teaches that failure is not the end of a journey, but a hint for how to succeed.

TEACHING SUMMARY

Getting Started - 20 minutes

1) Review
2) Vocabulary
3) Try, Try Again

Activity: Building a Foundation - 30 minutes

4) Building a Foundation

Wrap-up - 5 minutes

5) Flash Chat - What did we learn? 6) Vocab Shmocab

LESSON OBJECTIVES

Students will:

- 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 plan and structure until they satisfy challenge

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Building Elements
  - Gumdrops and Toothpicks (approx 30 each group) or
  - Marshmallows and Popsicle Sticks (approx 30 each group) or
  - Paper and Tape (approx 10 pieces per group)
- Mini Paper Cup for checking height (1 per group)

- Book for testing strength (1 per group)

For the Teacher

- [Lesson Video]
- Teacher Lesson Guide
- Print [Building a Foundation Packet]
- Gather enough building elements for each group
    - You don't have to give any certain amount; just make sure you put some limit on materials.

GETTING STARTED (20 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. You can do this as one large group or have students discuss with an elbow partner.

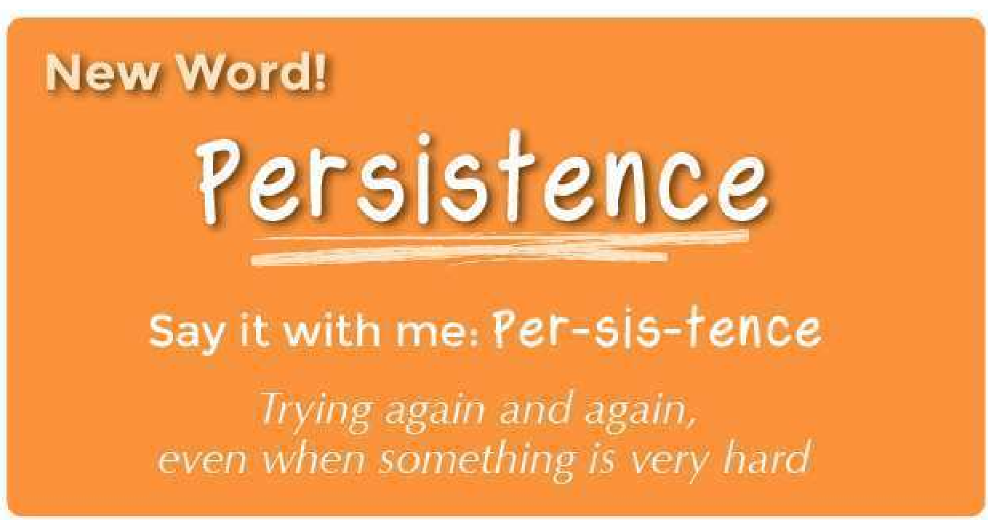
Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP *Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.*

2) 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

3) 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?

- Video of [baby learning to scoot, sit, walk]().
- 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

ACTIVITIES: (20 MIN)

4) Building a Foundation Activity

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:

1) Use only the supplies provided to build a tower.

2) The tower can be any shape, but it has to be at least as tall as the paper cup.

3) The tower must support the weight of a book for a full 10 seconds.

Directions:

1) Divide students into groups of 3 or 4.

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.

8) Congratulate the students as they succeed and take pictures of the successful towers (if possible) to upload to the Code.org site!

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.

WRAP-UP (5 MIN)

5) 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.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

"Going around the earth along the full circumference"
"Getting help from a large group of people to finish something faster"
"Trying again and again, even when something is very hard"

...and what is the word that we learned?

ASSESSMENT (0 MIN)

7) No Individual Assessment

- The final assessment of this lesson is the result of the previous activity.

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.

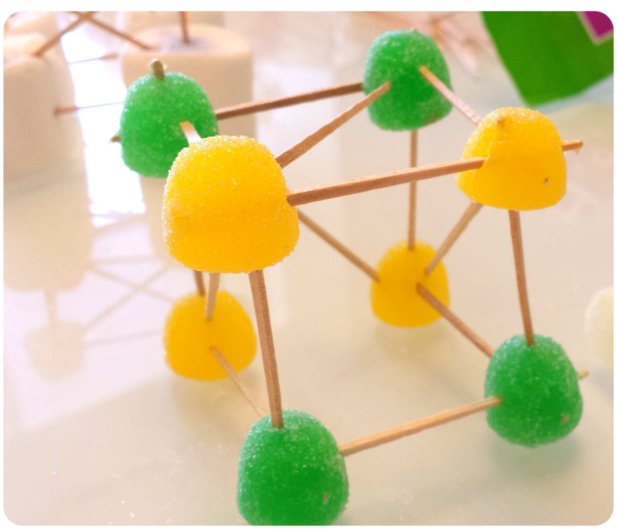
4) Challenge the class to think ahead to the problem and plan out their method of building their first tower.

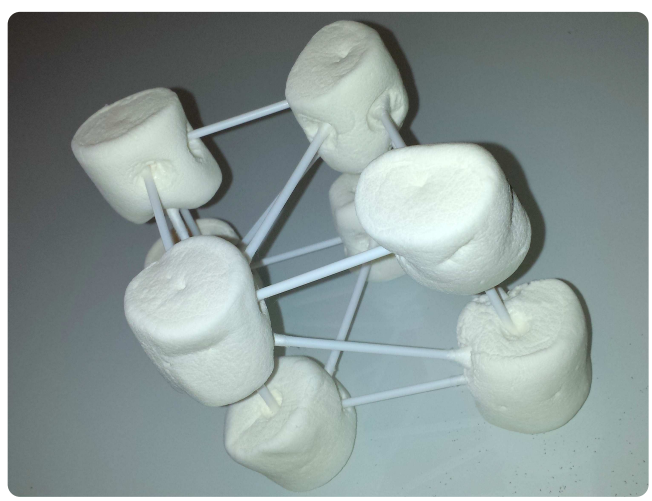
5) Encourage students to begin building, then have them alert you when they think they've met the challenge described by the rules.

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!





10

Course 1 | Lesson 10

Artist: Shapes

Lesson time: 30 Minutes

LESSON OVERVIEW

Returning to the Artist, the puzzles in this stage are focused on drawing, combining, and dividing squares and rectangles.

LESSON OBJECTIVES

Students will:

- Create a program to draw a shape using sequential steps
- Explain the difference between squares and rectangles and support it with evidence consisting of the commands used to draw the different shapes
- Compare the positions of different objects using "above," "below," "next to"
- Correctly identify shapes regardless of their overall size
- Compare and contrast squares and rectangles by their number of sides and side lengths

GETTING STARTED

Introduction

Before beginning this lesson, students should have a basic understanding of squares and rectangles.

- How do you make a square? A rectangle?
- What happens when you put two squares next to each other?

ACTIVITY

Artist: Shapes

If students struggle to use the correct number of blocks to draw a line, point out that each line segment has a dot on both ends.

EXTENDED LEARNING

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

The Copy Machine

- Give students two pieces of paper
- On one sheet have the students draw a simple image (right angles and straight lines only)
- On the second sheet draw instructions for recreating that image using a series of arrows
- Trade instruction sheets and attempt to recreate the image using only the provided instructions.

11

Course 1 | Lesson 11

Spelling Bee

Lesson time: 30 Minutes

LESSON OVERVIEW

Part puzzle, part word search, the Spelling Bee asks students to program a bee to find common words in a grid.

LESSON OBJECTIVES

Students will:

- Arrange sequential movement commands to search for and identify target words within a grid of letters.
- Practice spelling age-appropriate words

GETTING STARTED

Introduction

- Students should be able to read and identify the following words for this activity:
  - North
  - South
  - East
  - West
  - Jump
  - Code
  - Debug
  - Above
  - Below
  - Story
  - Move
  - Square

ACTIVITY

Spelling Bee

Very young students or struggling readers may need aditional support finding the words - using manipulatives (like Scrabble tiles) can help students see what words look like in different directions.

EXTENDED LEARNING

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

Vocab Hop Scotch

Using class vocab words, create a floor-sized word search. The whole class can then "program" a student, or teacher, to spell out words by creating sequences of cardinal directions.

U

UNPLUGGED

Getting Loopy

Lesson time: 15 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

Loops are a handy way of describing actions that repeat a certain numbers of times. In this lesson, students will practice converting sets of actions into a single loop.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) Repeat After Me

Activity: Loops - 15 minutes

4) Getting Loopy

Wrap-up - 10 minutes

5) Flash Chat - What did we learn?

Assessment - 10 minutes

6) Getting Loopy Assessment

LESSON OBJECTIVES

Students will:

- 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

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Open Space for Dancing/Moving
- Assessment Worksheet: Getting Loopy Assessment
- Pens/Pencils/Markers

For the Teacher

- Lesson Video

- Teacher Lesson Guide
- Print one Getting Loopy Activity Worksheet for the class
- Print Assessment Worksheet: Getting Loopy Assessment for each student

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. You can do this as one large group or have students discuss with an elbow partner.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP *Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.*

2) 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

3) 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 10 times, 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?

ACTIVITY: MIN)

4) Getting Loopy

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."

LESSON TIP

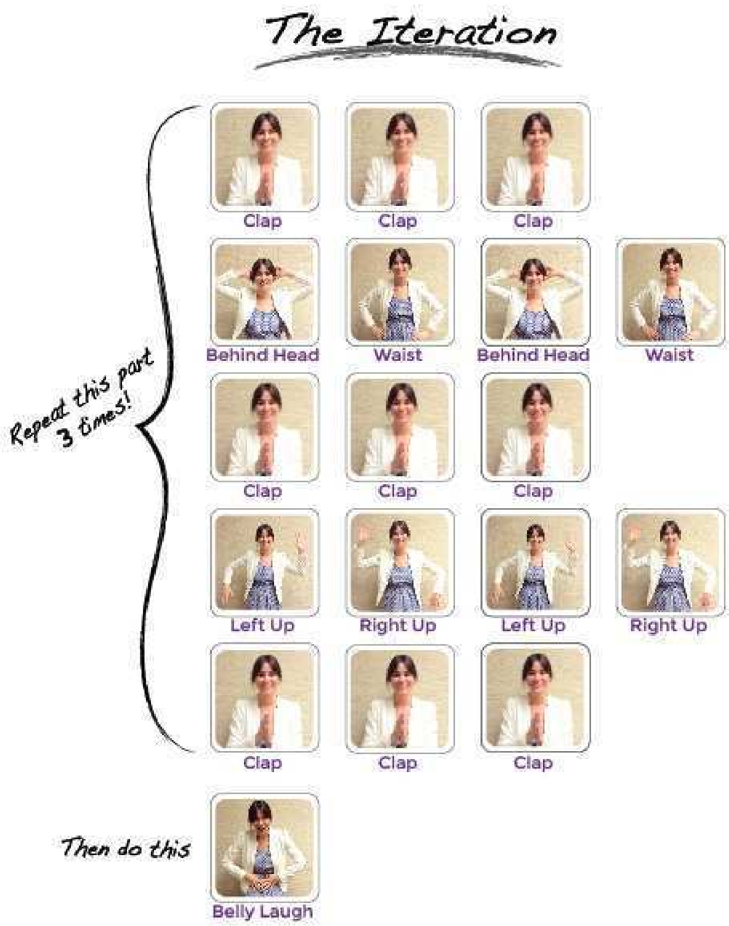
Looking for some good music? Here are some great places to find some:

- 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.

Directions:

1. Look at the dance moves provided on the Getting Loopy Worksheet.



2. Show the class what the entire dance looks like done at full-speed.

3. Run through the dance slowly, one instruction at a time, with the class.

4. 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?

5. Can you find anything else in the dance that we could use a loop for?

WRAP-UP (10 MIN)

5) 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?

ASSESSMENT (10 MIN)

6) Assessment Worksheet: Getting Loopy Assessment

- Hand out the worksheet titled "Getting Loopy" 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.

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

Name: ______________________ Date: ______________

Unplugged

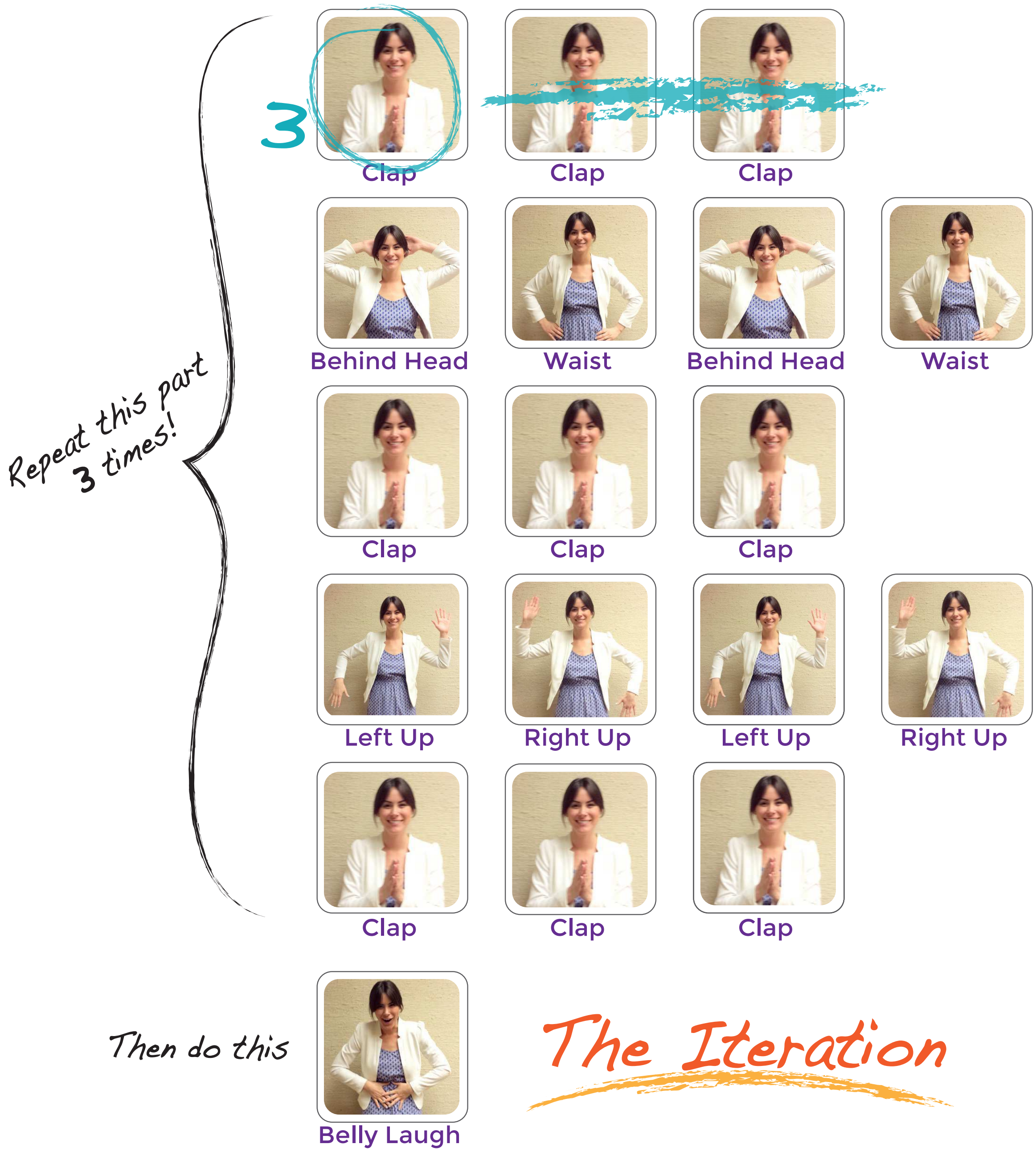
Getting Loopy

Unplugged Loops Activity

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.



13

Course 1 | Lesson 13

Maze: Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Building on the concept of repeating instructions from Getting Loopy, this stage will have students using loops to more efficiently traverse the maze.

LESSON OBJECTIVES

Students will:

- Identify the benefits of using a loop structure instead of manual repetition
- Create a program for a given task which loops a single command
- Break down a long sequence of instructions into the smallest repeatable sequence possible
- Create a program for a given task which loops a sequence of commands
- Employ a combination of sequential and looped commands to reach the end of a maze

GETTING STARTED

Introduction

Review with students the Getting Loopy activity:

- What are loops?
- Why do we use them?

ACTIVITY

Maze: Loops

As students work through the puzzles, see if they can figure out how many fewer blocks they use with a loop vs. not using a loop.

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!

14

Course 1 | Lesson 14

Bee: Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

In the last stage students used loops to repeat simple movements. Now they're going to add to that the looping of actions in order to help the bee collect more nectar and make more honey.

LESSON OBJECTIVES

Students will:

- Write a program for a given task which loops a single command
- Identify when a loop can be used to simplify a repetitive action
- Employ a combination of sequential and looped commands to move and perform actions

GETTING STARTED

Introduction

- What daily activites do you do that could be described as a loop?
- If you wanted someone to give you 5 cookies, how would you ask them to do it?
  - Push for a detailed descripton on this one. How can I take a cookie if the cookie jar is closed? How can I reach the cookie jar if I'm not in the kitchen?
- It's obviously far more efficient to do this activity as a loop, instead of laboriously giving all of the instructions for getting a cookie 5 times.
- We're going to help the bee be more efficient by looping the instructions to get lots of nectar and make lots of honey.

ACTIVITY

Bee: Loops

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!

U

UNPLUGGED

The Big Event

Lesson time: 15 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON 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.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) A Series of Events

Activity: Events - 15 minutes

4) The Big Event

Wrap-up - 5 minutes

5) Flash Chat - What did we learn?

Assessment - 10 minutes

6) The Big Event Assessment

LESSON OBJECTIVES

Students will:

- 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

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Assessment Worksheet: The Big Event Assessment
- Pens/Pencils/Markers

For the Teacher

- Lesson Video
- Teacher Lesson Guide

- Print one The Big Event Activity Worksheet and Event Controller
- Print Assessment Worksheet: The Big Event Assessment for each student

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. You can do this as one large group or have students discuss with an elbow partner.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) Vocabulary

This lesson has one new and important vocabulary word:

Event - Say it with me: E-vent
An action that causes something to happen

3) 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.

ACTIVITY: (15 MIN)

4) The Big Event

- Do you remember helping the Flurbs find fruit? How about making maps where you helped your friend get to the smiley face?
  - In those exercises, you knew in advance exactly where you wanted your character 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.

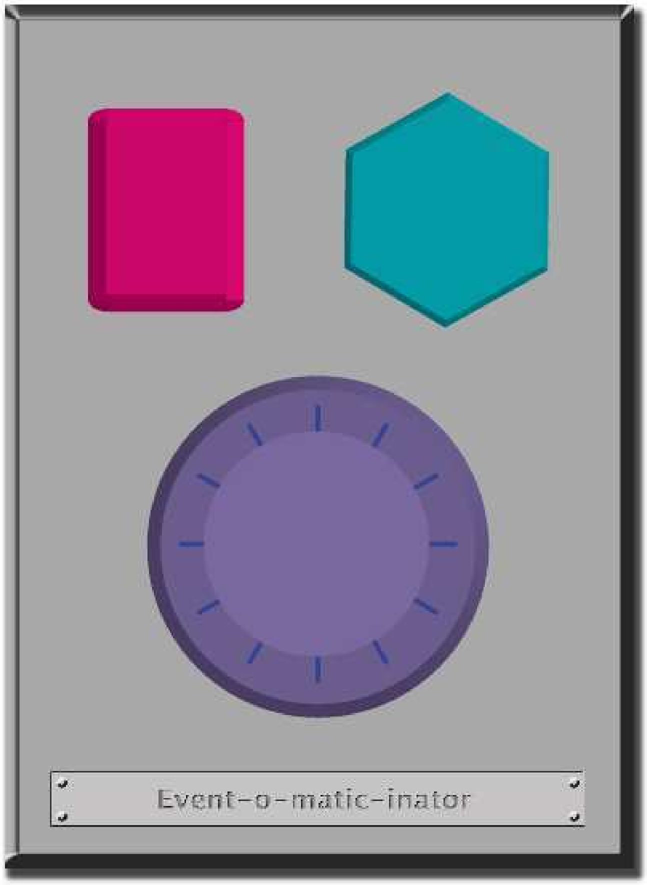
In earlier lessons, we created algorithms that allowed us to control a friend or Flurb for several steps at a time. It was fun and useful, but what happens when you don't know everything that you want your friend to do in advance? This is where events come in!

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.

Directions:

1. Project the Event Controller onto your classroom screen.



1. Decide with your class what each button does. We suggest:
   - Pink Button -> Say "Wooooo!"
   - Teal Button -> "Yeah!"
   - Purple Dial -> "Boom!"
2. Practice tapping the buttons on the overhead and having your class react.
3. Add some button sequences into the mix and have the students try to keep up with their sounds.
4. 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.
5. Get the class started on a planned task before interrupting them again with the buttons. We suggest:
   - Counting to 10
   - Singing "Old MacDonald"
6. Once their plan is underway, interject button presses sporadically.
7. 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)

5) 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)

6) Assessment Worksheet: Controlling by Events 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

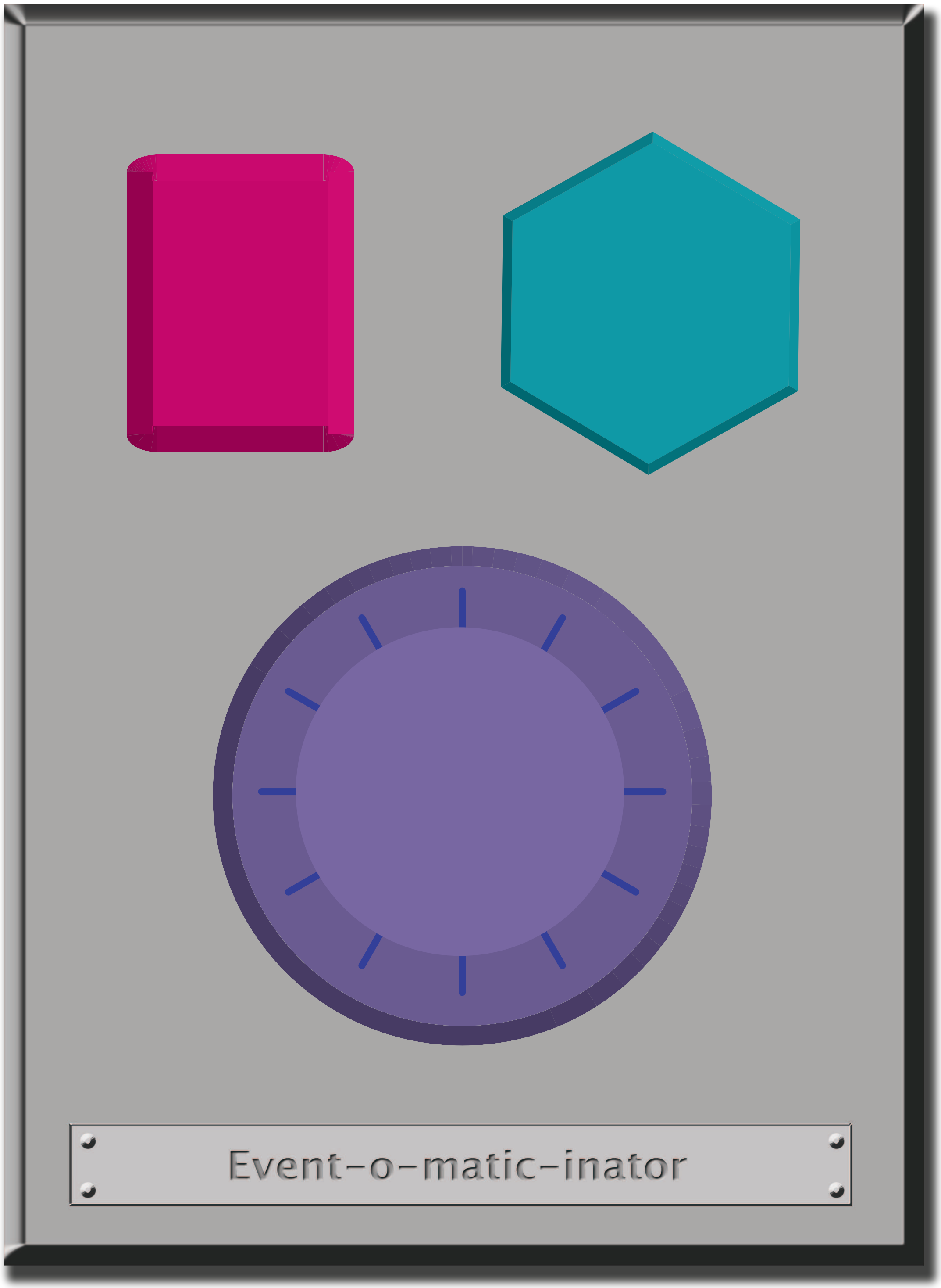
Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

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

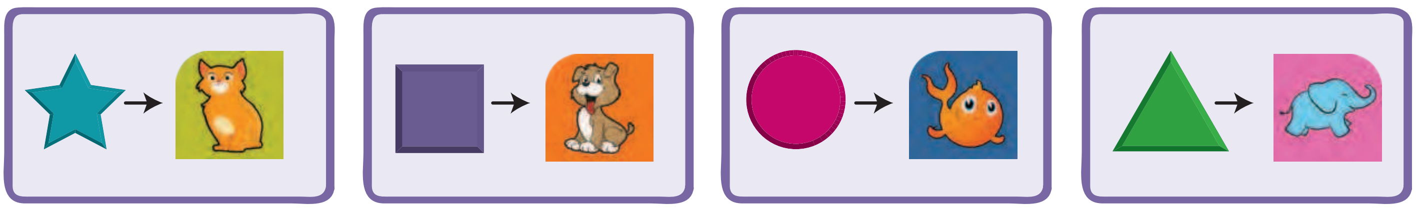
Name: ______________________ Date: ______________

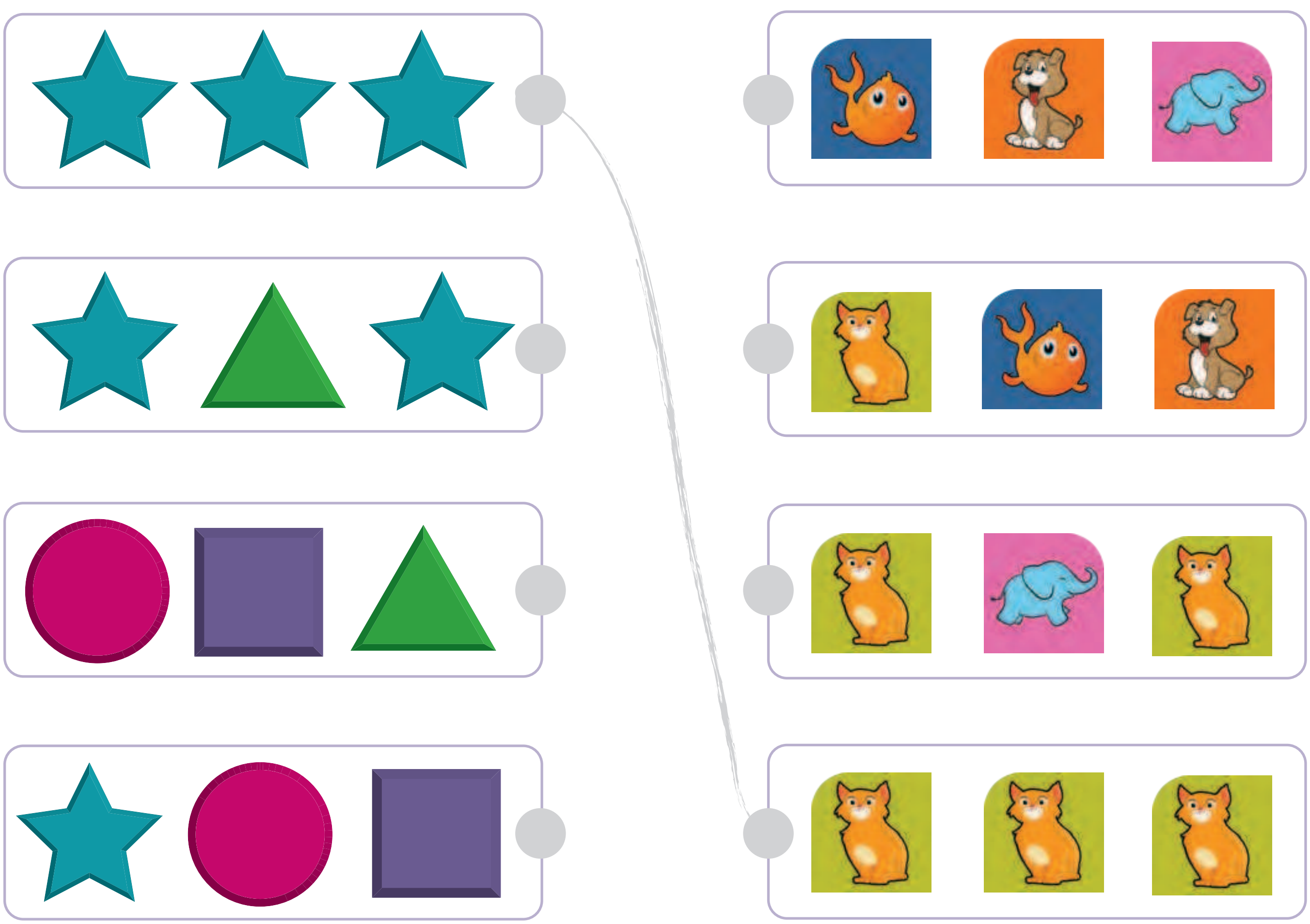
The Big Event

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.





16

Course 1 | Lesson 16

Play Lab: Create a Story

Lesson time: 30 Minutes

LESSON OVERVIEW

In this culminating plugged activity, students will have the opportunity to apply all of the coding skills they've learned to create an animated story. It's time to get creative and create a story in the Play Lab!

LESSON OBJECTIVES

Students will:

- Identify actions that correlate to input events
- Create an animated, interactive story using sequence, loops, and event-handlers
- Share a creative artifact with other students

GETTING STARTED

Introduction

- Review The Big Event activity with students:
  - What did we "program" the button click events to do?
- Now we're going to add events to our coding, Specifically, we're going to have an event for when two characters touch each other.
  - In video game programming we call this kind of event collision detection; it lets us decide what to do when one thing collides with, or touches, another.
  - What kinds of collision events have you seen in games?

LESSON TIP

Students will have the opportunity to share their final product with a link. This is a great opportunity to show your school community the great things your students are doing. Collect all of the links and keep them on your class website for all to see!

ACTIVITY

Play Lab: Create a Story

This is the most free-form plugged activity of the course. At the final stage students have the freedom to create a story of their own. You may want to provide structured guidelines around what kind of story to write, partiularly for students who are overwhelmed by too many options.

EXTENDED LEARNING

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

Look Under the Hood

When you share a link to your story, you also share all of the code that goes behind it. This is a great way for students to learn from each other.

- Post links to completed stories online or on the board.
  - Make a story of your own to share as well!
- When students load up a link, have them click the "How it Works" button to see the code behind the story.
- Discuss as a group the different ways your classmates coded their stories.
  - What suprised you?
  - What would you like to try?
- Choose someone else's story and build on it. (Don't worry, the original story will be safe.)

U

UNPLUGGED

Going Places Online

Lesson time: 30 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

In collaboration with Common Sense Media 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.

TEACHING SUMMARY

Getting Started - 20 minutes

1) Review 2) Vocabulary
3) Where We Go

Activity: Keep It Private - 30 minutes

4) Keep It Private

Wrap-up - 10 minutes

5) Flash Chat - What did we learn? 6) Vocab Shmocab

Assessment - 5 minutes

7) Keep It Private Assessment

LESSON OBJECTIVES

Students will:

- Understand that being safe when they visit websites is similar to staying safe in real life
- Learn to recognize websites that are alright for them to visit
- Recognize the kind of information that is private.
- Understand that they should never give out private information on the Internet
- Learn to create effective usernames that protect their private information

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Pens, Pencils and Paper
- Keep It Private Assessment

For the Teacher

- Lesson Video
- Teacher Lesson Guide
- Download or prepare the "My Online Neighborhood" 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

GETTING STARTED (20 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. You can do this as one large group or have students discuss with an elbow partner.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) 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”

3) 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 My Online Neighborhood 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.

ACTIVITIES: (20 MIN)

4) 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:

1) Ask a parent or other trusted adult before you create a username. 2) 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. 3) Avoid using symbols or spaces, as they are usually not allowed in usernames.

Tips

1) 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.

2) If the username you create is already taken, you will have to come up with another one.

3) 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 Media page on Scope and Sequence.

WRAP-UP (5 MIN)

5) 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	Inappropriate
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.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

"Traveling at the speed of light"
"A name you make up so that you can see or do things on a website, sometimes called a 'screen name'" "A digital environment that looks real"

...and what is the word that we learned?

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 Media

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

Name: ______________________ Date: ______________

U

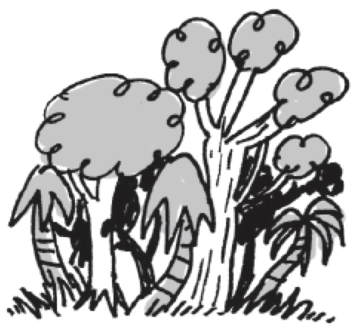
Unplugged

Keep It Private

Learning to be Safe and Responsible Assessment

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

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

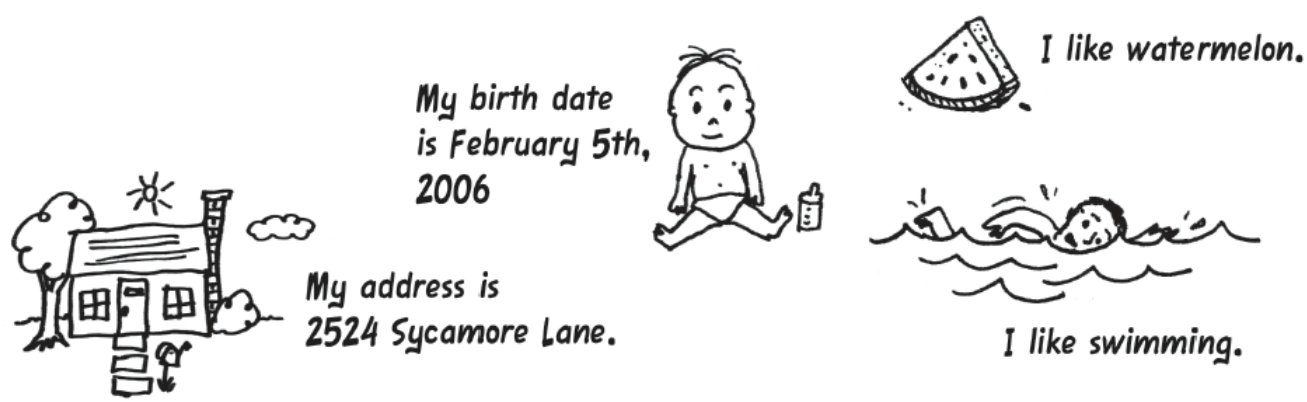


THE JUNGLE

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.



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

18

Course 1 | Lesson 18

Artist: Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Returning to the artist, students learn to draw more complex images by looping simple sequences of instructions.

LESSON OBJECTIVES

Students will:

- Count the number of times an action should be repeated and represent it as a loop
- Decompose a shape into its smallest repeatable sequence
- Create a program that draws complex shapes by repeating simple sequences

GETTING STARTED

Introduction

- Ask students to name as many simple shapes as possible, focus on shapes with equal sides and angles.
- For each shape:
  - How would you explain to someone how to draw that shape?
  - How could you draw this using a loop?

ACTIVITY

Artist: Loops

In the Artist levels students will no longer be constrained to 90 degree angles. Having protractors available can be help students better visualize the angles they need.

EXTENDED LEARNING

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

The Copy Machine

- Give students two pieces of paper.
- On one sheet have the students draw a shape with equal sides and angles.
- On the second sheet draw instructions for recreating that shape using loops.
- Trade instruction sheets and attempt to recreate the shape using only the provided instructions.
  - Can you predict what shape will be drawn just by reading the instructions?

Course 2

OVERVIEW

Students create programs with loops, events, and conditionals and write algorithms for everyday tasks. They will translate their names into binary, investigate different problem-solving techniques, and discuss societal impacts of computing. By the end of the curriculum, students create interactive games or stories they can share. While the description of some lessons may look similar to lessons in the Course 1, this review is important for those who have taken Course 1 as most will be at the lower elementary level. The complexity and depth of topics discussed are scaffolded appropriately to provide all students a rich and novel experience. Students starting in Course 2 will be students who can read in the lower and middle elementary grades.

Lesson Sequence

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

#	Lesson Name	Description
1	**Graph Paper Programming**	Students write an algorithm (a set of instructions) using a set of predefined commands to direct their classmates to reproduce a drawing.
2	**Real-Life Algorithms**	This lesson calls out ways we use algorithms in our daily lives. This lesson also focuses on the bigger picture of computer science and how algorithms play an essential part.
3	Maze: Sequence	Students write programs (algorithms for the computer) that get a character through a maze. They'll understand the importance of sequence in the programs they write.
4	Artist: Sequence	Students write programs to draw different lines and shapes.
5	**Getting Loopy**	This lesson introduces the programming concept of loops (repeated instructions) through a dance activity. Students will learn simple choreography and then be instructed to repeat it.
6	Maze: Loops	Student write programs in the Maze environment using loops.
7	Artist: Loops	Students write programs to draw different shapes while identifying patterns in their code. They learn about the programming concept of loops (repeated statements), which can be used to make their programs more efficient.
8	Bee: Loops	Students write programs using loops in the Bee environment.

#	Lesson Name	Description
9	**Relay programming**	Students run a relay race, where they dash across the yard to write an algorithm based on a "Graph Paper Programming" image. They can only write one instruction at a time and if there's an error, they have to erase everything back to the error.
10	Bee: Debugging	Students are presented with a pre-written program that fails to complete the puzzle. Students will have to “debug” or fix the pre-written program.
11	Artist: Debugging	Students are presented with a drawing and a pre-written program that fails to create that drawing. Students will have to “debug” or fix the pre-written program.
12	**Conditionals**	To learn about conditional statements, students play a card game and create rules like “If I draw a red card, I get a point” and “If I draw a black card, you get a point.”
13	Bee: Conditionals	Students write programs using conditional statements using the Bee environment.
14	**Binary Bracelets**	Students create bracelets from a paper template that is a binary representations of the first letter of their name. Students learn that the same set of data can be represented in more than one way.
15	**The Big Event**	Students are introduced to the programming concept of “events,” which are actions that a computer constantly monitors for. The teacher will press buttons on a fake remote, and student have to shout specific phases depending on which button is pressed.
16	Flappy	Using the concept of “Events,” students will create their own game with events like “When the mouse is clicked, make the bird flap” and “When the bird hits the ground, end the game.”
17	Play Lab: Create a Story	Students employ all the different programming concepts they have learned in the curriculum this far to make a customized, interactive story or game of their own.
18	**Your Digital Footprint**	Teachers introduce to students the idea that putting information about themselves online creates a digital footprint or “trail” that has consequences.
19	Artist: Nested Loops	Students write programs that draw interesting and beautiful patterns using nested loops.

U

UNPLUGGED

Graph Paper Programming

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

By "programming" one another to draw pictures, students will begin to understand what programming is really about. The class will begin by having students instruct each other to color squares in 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.

TEACHING SUMMARY

LESSON OBJECTIVES

Students will:

- Understand the difficulty of translating real problems into programs
- Learn that ideas may feel clear and yet still be misinterpreted by a computer
- Practice communicating ideas through codes and symbols

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Four-by-Fours Activity Worksheet
- Graph Paper Programming Assessment
- Sheets of 4x4 paper grids for the students to use as practice (These are provided as part of the Four-by-Fours

Activity Worksheet, but if you have the students create their own, you can include Common Core Math standard 2.G.2.)
- Blank paper or index cards for programs
- Markers, pens, or pencils

For the Teacher

- Lesson Video
- Print out one Four-by-Fours Activity Worksheet for each group
- Print one Graph Paper Programming Assessment for each student
- Supply each group with several drawing grids, paper, and pens/pencils

GETTING STARTED (15 MIN)

1) 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

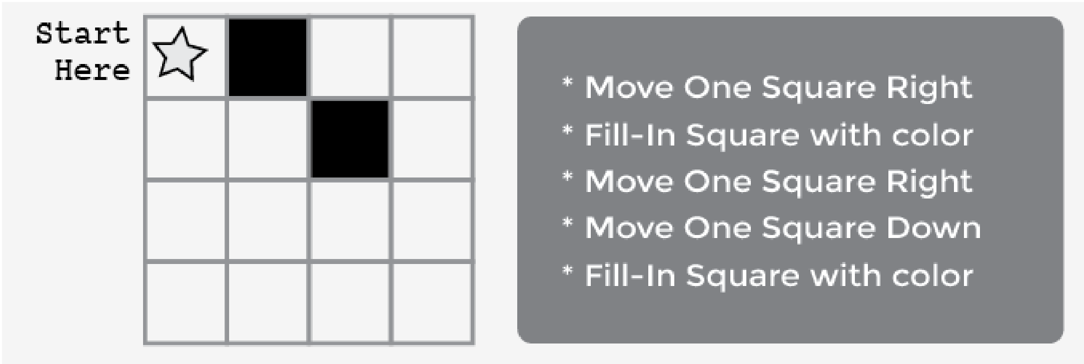
2) Introduce Graph Paper Programming

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.

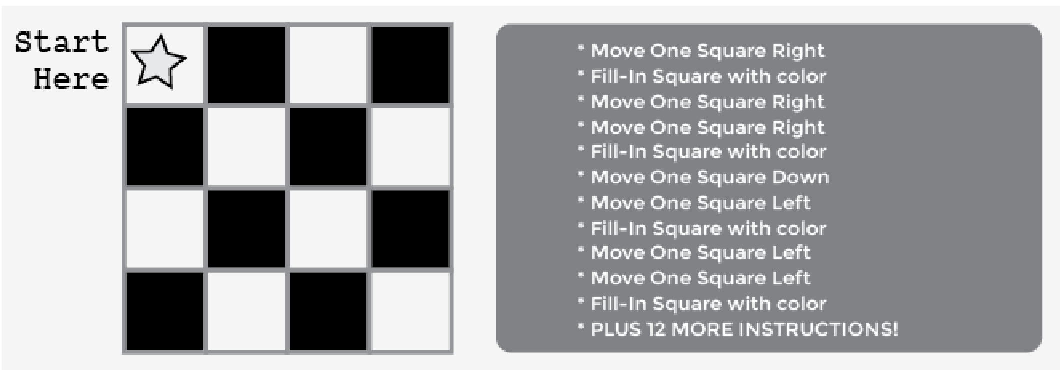
For this exercise, we will use sheets of 4x4 graph paper. Starting at the upper left-hand corner, we'll guide our teammates' Automatic Realization Machine (ARM) with simple instructions. Those instructions include:

- 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:



That's simple enough, but it would take a lot of writing to provide instructions for a square like this:

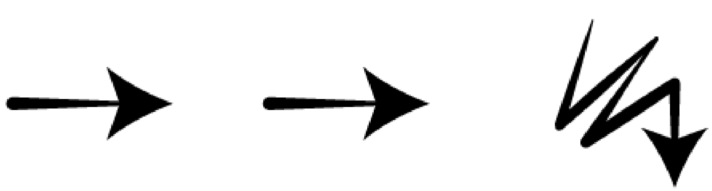


With one little substitution, we can do this much more easily! Instead of having to write out an entire phrase for each instruction, we can use arrows.

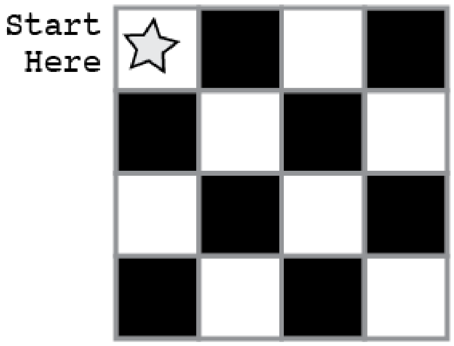
In this instance, the arrow symbols are the "program" code and the words are the "algorithm" piece. This means that we could write the algorithm:

"Move one square right, Move one square right, Fill-in square with color"

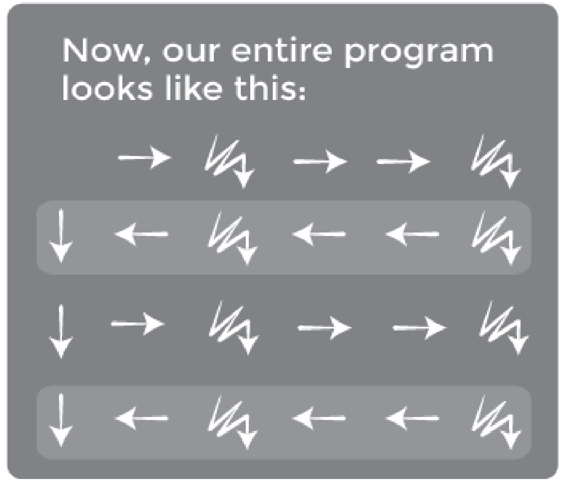
and that would correspond to the program:



Using arrows, we can redo the code from the previous image much more easily!



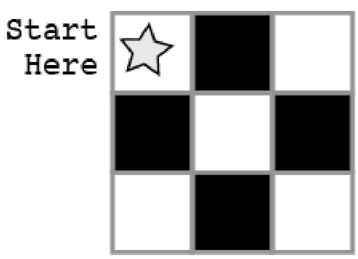
Follow along with your finger and see if you can figure out how to get this image from the program to the right.



3) 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.

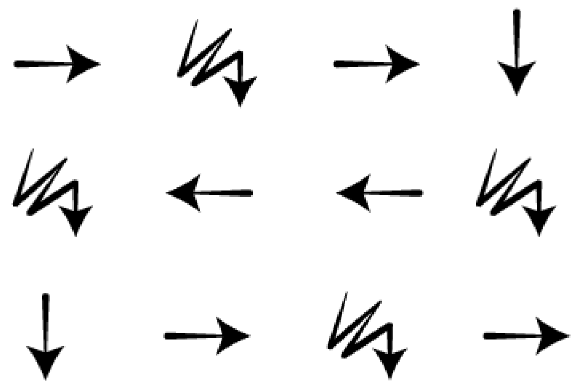


This is a good way to introduce all of the symbols in the key. To begin, fill in the graph for the class -- square by square -- then ask them to help describe what you've just done. First, you can speak the algorithm out loud, then you can turn your verbal instructions into a program.

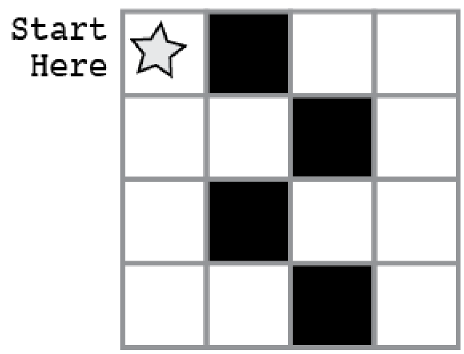
A sample algorithm:

> "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 Four-by-Fours Activity Worksheet.

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.

ACTIVITY: GRAPH PAPER PROGRAMMING (20 MIN)

4) Four-by-Fours Activity Worksheet

1. Divide students into pairs.
2. Have each pair choose an image from the worksheet.
3. Discuss the algorithm to draw that image with partner.
4. Convert algorithm into a program using symbols.
5. Trade programs with another pair and draw one another's image.
6. Choose another image and go again!

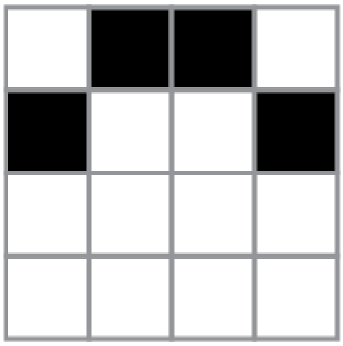


Image 1

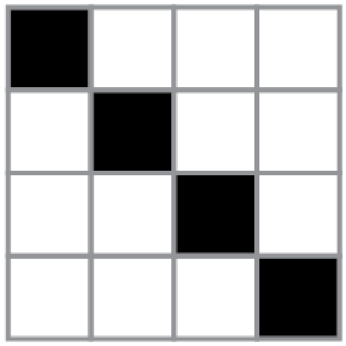


Image 2

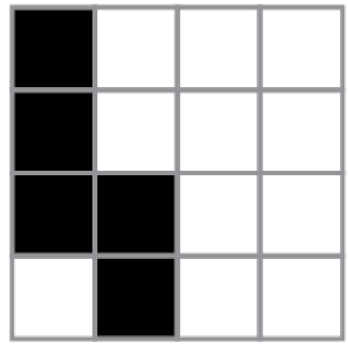


Image 3

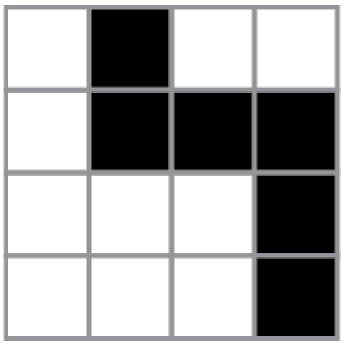


Image 4

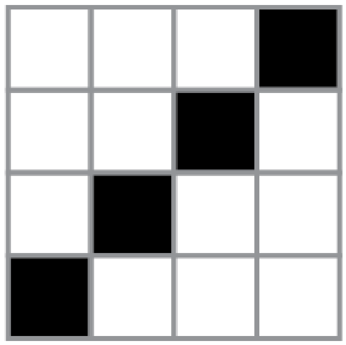


Image 5

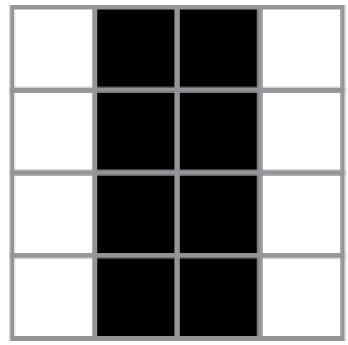


Image 6

WRAP-UP (5 MIN)

5) 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?

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

    "A large tropical parrot with a very long tail and beautiful feathers"
    "A list of steps that you can follow to finish a task"
    "An incredibly stinky flower that blooms only once a year"

    ...and what is the word that we learned?

- Which one of these is the *most* like a "program"?

    *A shoebox full of pretty rocks
    *Twelve pink flowers in a vase
    *Sheet music for your favorite song

    Explain why you chose your answer.

ASSESSMENT (10 MIN)

7) Graph Paper Programming Assessment

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

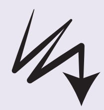
Graph Paper Programming

Four-by-Fours Activity Worksheet

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

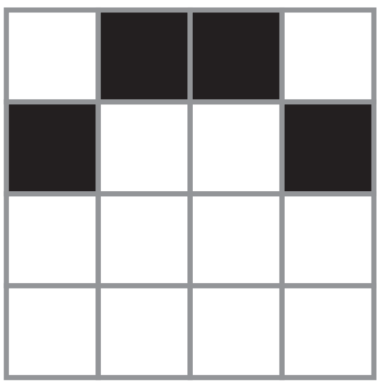


Image 1

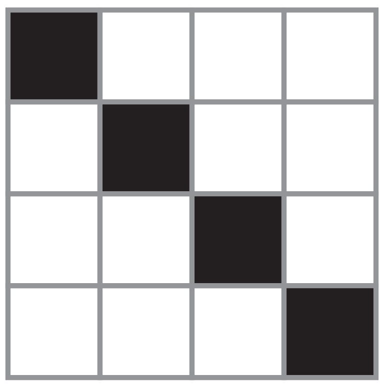


Image 2

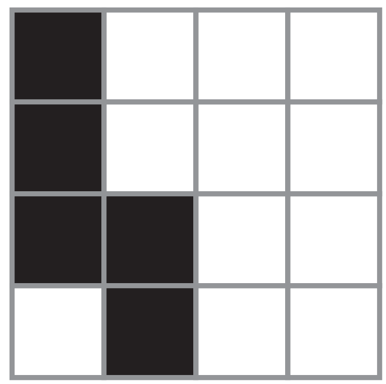


Image 3

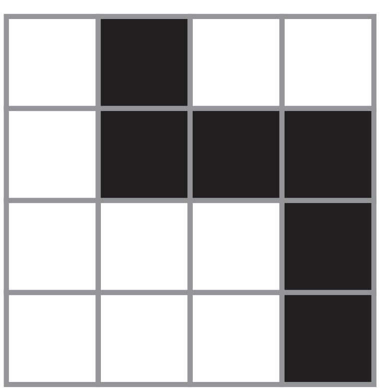


Image 4

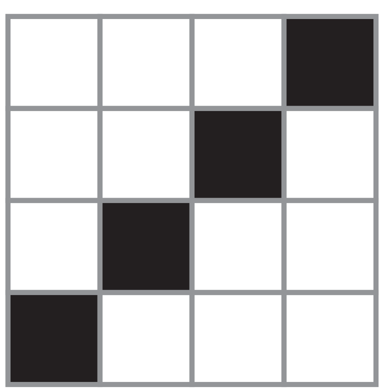


Image 5

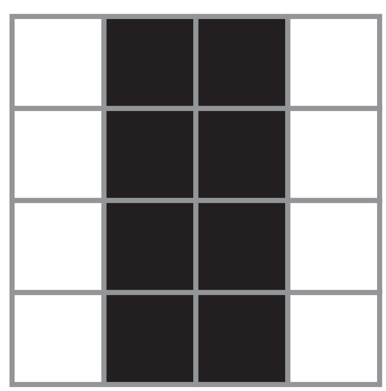


Image 6

Name: ______________________ Date: _______________

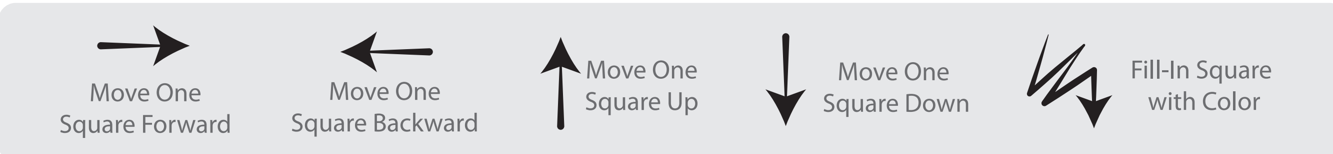
Unplugged

Graph Paper Programming

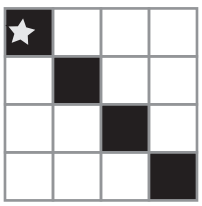
Assessment Worksheet

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.

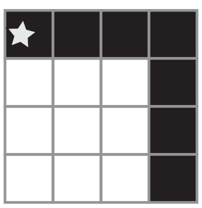


Start Here



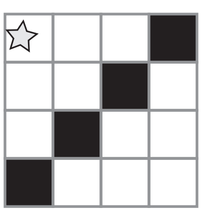
Step 1	2	3	4	5	6	7	8	9	10
Step 11	12	13	14	15	16	17	18	19	20

Start Here



Step 1	2	3	4	5	6	7	8	9	10
Step 11	12	13	14	15	16	17	18	19	20

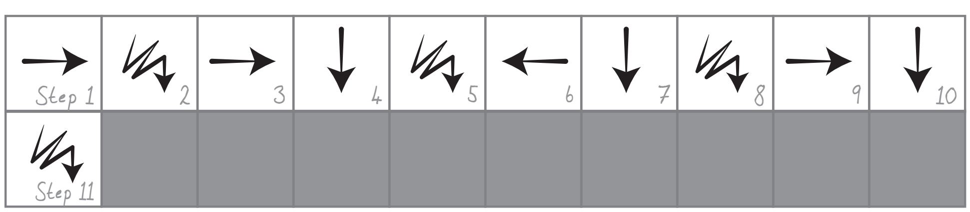
Start Here



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



U

UNPLUGGED

Real-Life Algorithms: Paper Airplanes

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

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

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) What We Do Daily

Activity: Real-Life Algorithms - 20 minutes

4) Real-Life Algorithms: Paper Airplanes

Wrap-up - 5 minutes

5) Flash Chat - What did we learn?

Assessment - 10 minutes

6) Daily Algorithms

LESSON OBJECTIVES

Students will:

- Name various activities that make up their day
- Decompose large activities into a series of smaller events
- Arrange sequential events into their logical order

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Paper for folding into airplane
- Real-Life Algorithms Worksheet: Paper Airplanes

- Assessment Worksheet: Daily Algorithms
- Scissors
- Glue

For the Teacher

- Lesson Video
- Teacher Lesson Guide
- Real-Life Algorithms Worksheet: Paper Airplanes
- Print Assessment Worksheet: Daily Algorithms for each student

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) 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

3) 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!

ACTIVITY: (20 MIN)

4) Real-Life Algorithm Worksheet: Paper Airplanes

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.*

- 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.

Directions:

1. Cut out the steps for making a paper airplane provided worksheet.
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!
5. 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.

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.*

WRAP-UP (5 MIN)

5) Flash Chat: What did we learn?

- How many of you were able to follow your classmates' algorithms to make your airplanes?
- 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: "Fold a Paper Airplane"?
        - Would it have been easier or harder?
        - What if it were forty steps?
- What was your favorite part about that activity?

ASSESSMENT (15 MIN)

6) Assessment Worksheet: Daily Algorithms

- Hand out the worksheet titled "Daily Algorithms" 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: ______________

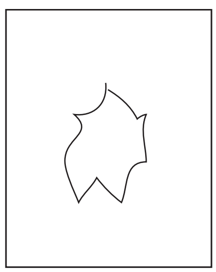
Unplugged

Real-Life Algorithms

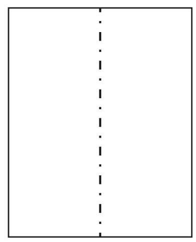
Paper Airplane Worksheet

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!

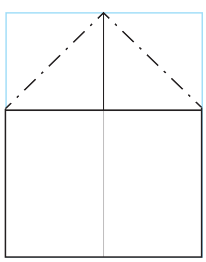


CUT CENTER OUT OF PAPER

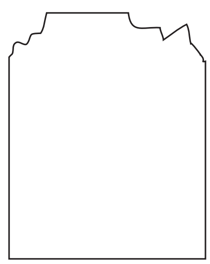


CREASE PAPER DOWN THE CENTER

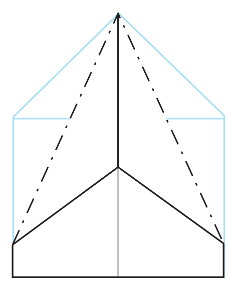
CRUMBLE PAPER



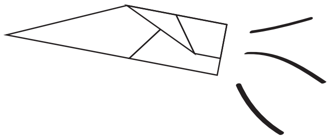
FOLD TOP CORNERS TO CENTER



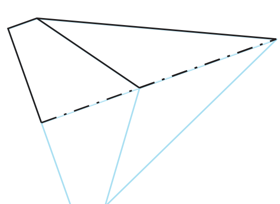
RIP CORNER OFF PAPER



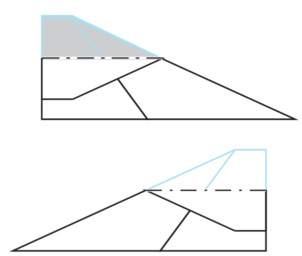
FOLD CORNER SIDES TO CENTER



TOSS FINISHED PLANE



FOLD PAPER IN HALF AGAIN



PULL SIDES DOWN

Name: ______________________ Date: ________________

Unplugged

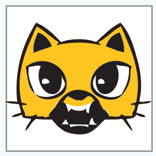
Daily Algorithms

Assessment Worksheet

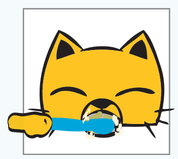
An algorithm is a list of instructions for accomplishing a task. We follow algorithms everyday when it comes to activities like making the bed, making breakfast, or even getting dressed in the morning.

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.

Teeth are clean!

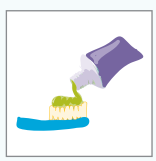


____



____

____



____

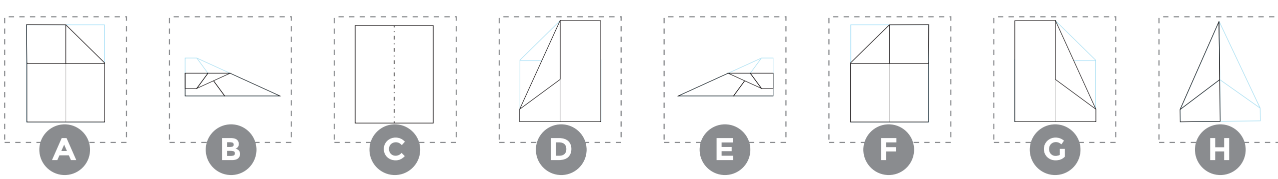
Step 1

Step 2

Step 3

Step 4

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: ____ ____ ____ ____ ____ ____ ____ ____

3

Course 2 | Lesson 3

Maze: Sequence

Lesson time: 30 Minutes

LESSON OVERVIEW

In this series of puzzles students will build on the understanding of algorithms learned in the Graph Paper Programming and Real-Life Algorithms Unplugged activities. Featuring characters from the game Angry Birds, students will develop sequential algorithms to move a bird from one side of the maze to the pig at the other side. To do this they will stack blocks together in a linear sequence to move straight or turn left and right.

LESSON OBJECTIVES

Students will:

- Express movement as a series of commands
- Order movement commands as sequential steps in a program
- Represent an algorithm as a computer program
- Count the number of times an action should be executed and represent it as instructions in a program
- Recall and apply the rules of pair programming
- Use pair programming to complete collaborative tasks with or without a computer
- Identify situations when the rules of pair programming are not followed

GETTING STARTED

Introduction

Ask your students if they are familiar with the game Angry Birds. Explain that they will be writing programs to help an Angry Bird locate a Pig.

- Getting the bird to the pig will require putting your directions in a very specific order or sequence.
- Can you solve the puzzles using the fewest blocks possible?

LESSON TIP

Some students may struggle with turning their bird in the correct direction, particularly when the bird isn't facing up. Remind students that when we say turn left or right, we're talking about it from the bird's point of view.

ACTIVITY

Maze: Sequence

As your students work through the puzzles, observe how they plan the path for the bird. Identify different strategies used and ask students to share with the whole class. This helps students to recognize that there are many ways to approach these problems. You may want to go through a few puzzles on the projector. While doing this you can ask one student to trace the path on the screen while another writes the directions on a whiteboard.

EXTENDED LEARNING

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

Create Your Own

In small groups, let students design their own mazes and challenge each other to write programs to solve them. For added fun, make life-size mazes with students as the pig and bird.

4

Course 2 | Lesson 4

Artist: Sequence

Lesson time: 30 Minutes

LESSON OVERVIEW

In this lesson students will take control of the Artist to complete simple drawings on the screen.

LESSON OBJECTIVES

Students will:

- Create a program to complete an image using sequential steps
- Select an argument for a given command
- Differentiate between defining and non-defining attributes of triangles, squares, and rectangles
- Draw triangles, squares, and rectangles to reflect defining attributes
- Explain the difference between squares and rectangles and support it with evidence consisting of the commands used to draw the different shapes
- Compare and contrast squares and rectangles by their number of sides and side lengths
- Compose two-dimensional shapes (rectangles, squares, trapezoids, triangles) to create a composite shape, such as two squares to compose a rectangle and two rectangles to compose a square
- Compose new shapes from composite shapes
- Draw partitions into a rectangle and describe the partitions using the words halves, fourths, quarters, half of, fourth of, and quarter of
- Describe a whole rectangle as two halves or four quarters
- Explain that decomposing into more equal shares creates smaller shares

GETTING STARTED

Introduction

Brainstorm with students ways to tell someone else how to draw a picture:

- How would you do that with a computer?
- In these puzzles you will be moving a character who leaves a line everywhere it goes.

ACTIVITY

Artist: Sequence

In the Artist levels students will no longer be constrained to 90 degree angles. Having protractors available can be help students better visualize the angles they need.

EXTENDED LEARNING

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

The Copy Machine

- Give students two pieces of paper
- On one sheet draw a simple image, using straight lines only.
- On the second sheet draw instructions for recreating that image commands to move straight and turn at various angles.
- Trade instruction sheets and attempt to recreate the image using only the provided instructions.

U

UNPLUGGED

Getting Loopy

Lesson time: 15 Minutes   Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

Loops are a handy way of describing actions that repeat a certain number of times. In this lesson, students will practice converting sets of actions into a single loop.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) Repeat After Me

Activity: Loops - 15 minutes

4) Getting Loopy

Wrap-up - 10 minutes

5) Flash Chat - What did we learn?

Assessment - 10 minutes

6) Getting Loopy Assessment

LESSON OBJECTIVES

Students will:

- 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

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Open Space for Dancing/Moving
- Assessment Worksheet: Getting Loopy Assessment
- Pens/Pencils/Markers

For the Teacher

- Lesson Video

- Teacher Lesson Guide
- Print one [Getting Loopy Activity Worksheet]() for the class
- Print Assessment Worksheet: [Getting Loopy Assessment]() for each student

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) 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

3) 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 10 times, 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?

ACTIVITY: (15 MIN)

4) Getting Loopy

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."

LESSON TIP

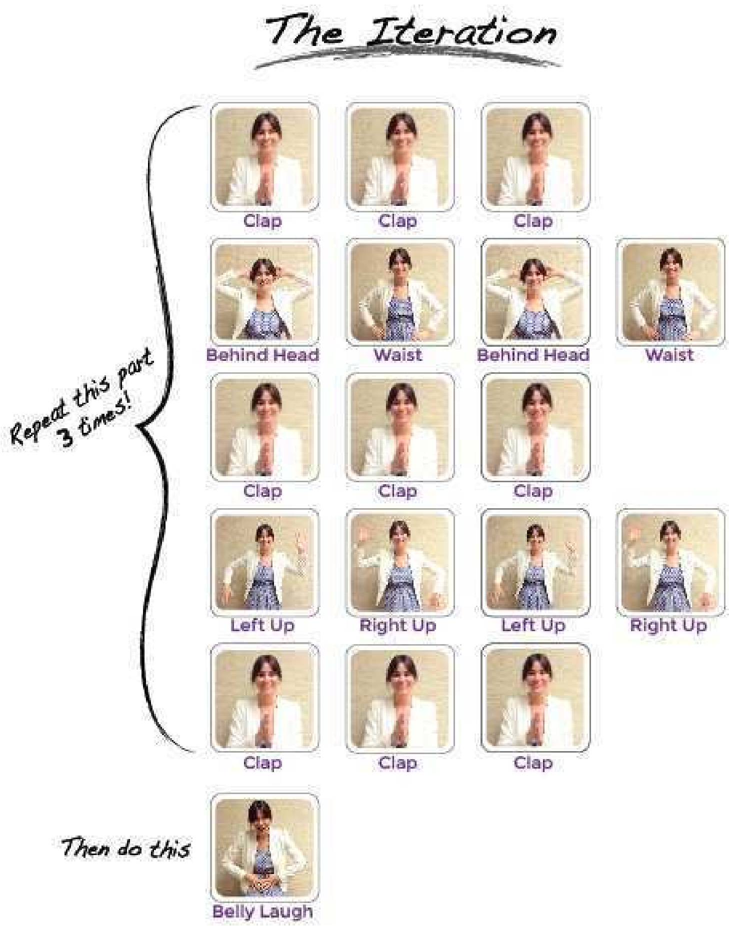
Looking for some good music? Here are some great places to find some:

- 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.

Directions:

1. Look at the dance moves provided on the Getting Loopy Worksheet.



2. Show the class what the entire dance looks like done at full-speed.
3. Run through the dance slowly, one instruction at a time, with the class.
4. 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?
5. Can you find anything else in the dance that we could use a loop for?

WRAP-UP (10 MIN)

5) 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?

ASSESSMENT (10 MIN)

6) Assessment Worksheet: [Getting Loopy Assessment]

- Hand out the worksheet titled "Getting Loopy" 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.

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!

Name: ______________________ Date: ______________

U

Unplugged

Getting Loopy

Unplugged Loops Activity

CODE

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."

We are going to practice using loops to explain how many times we should perform an action while we learn a new dance!

Come on everybody, let's do The Iteration!

New Word!

Loop

Say it with me: Loop

The action of doing something over and over again.

When you repeat something multiple times, like clapping your hands, you are perfoming a **loop** of that action.

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

Name: ____________________ Date: ______________

U

Unplugged

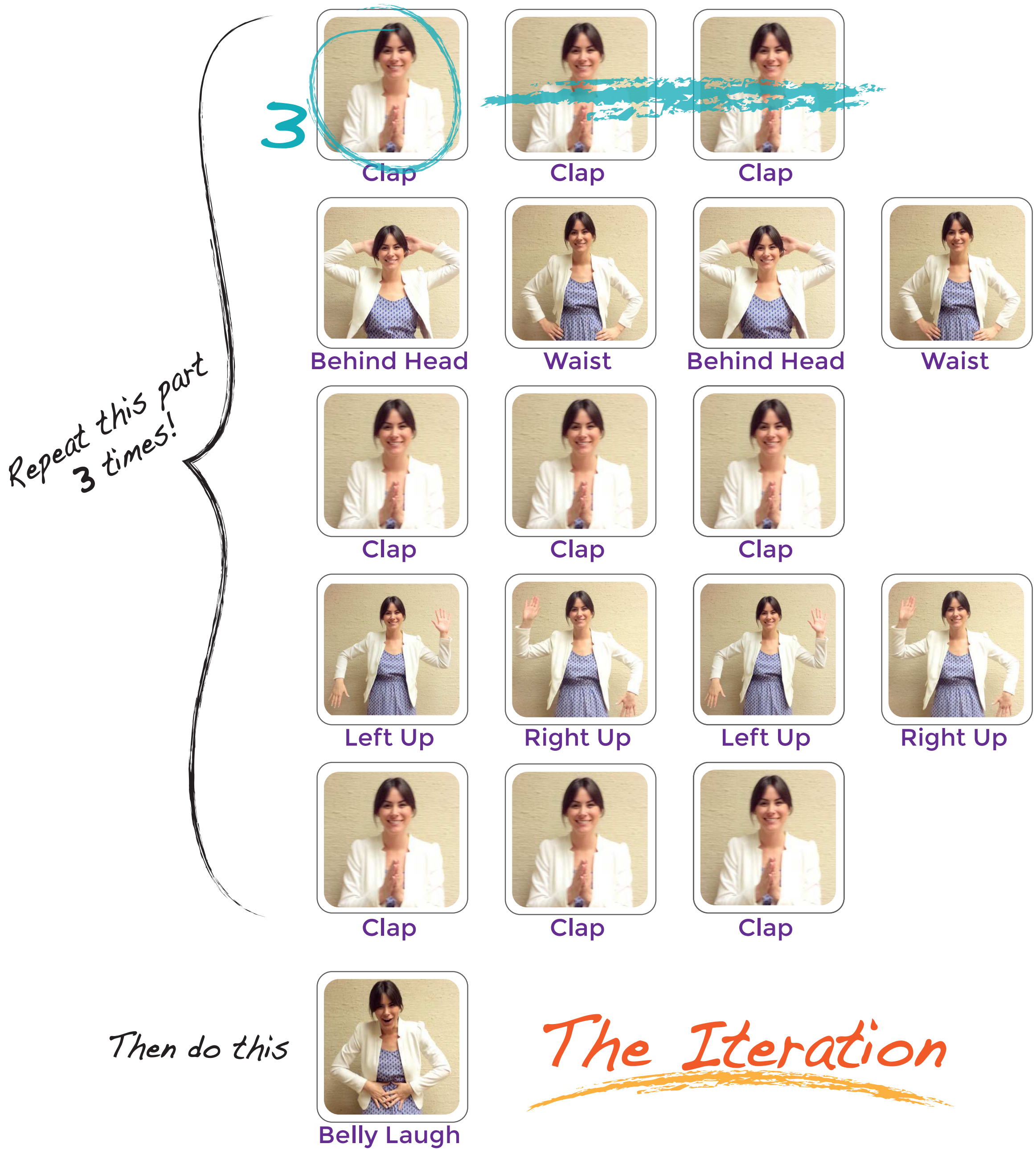
Getting Loopy

Unplugged Loops Activity

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.



6

Course 2 | Lesson 6

Maze: Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Building on the concept of repeating instructions from Getting Loopy, this stage will have students using loops to more effeciently traverse the maze.

LESSON OBJECTIVES

Students will:

- Identify the benefits of using a loop structure instead of manual repetition
- Create a program for a given task which loops a single command
- Break down a long sequence of instructions into the smallest repeatable sequence possible
- Create a program for a given task which loops a sequence of commands
- Employ a combination of sequential and looped commands to reach the end of a maze

GETTING STARTED

Introduction

Review with students the Getting Loopy activity:

- What are loops?
- Why do we use them?

ACTIVITY

[Maze: Loops](#)

As students work through the puzzles, see if they can figure out how many fewer blocks they use with a loop vs. not using a loop.

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!

7

Course 2 | Lesson 7

Artist: Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Returning to the artist, students learn to draw more complex images by looping simple sequences of instructions.

LESSON OBJECTIVES

Students will:

- Count the number of times an action should be repeated and represent it as a loop
- Decompose a shape into its smallest repeatable sequence
- Create a program that draws complex shapes by repeating simple sequences

GETTING STARTED

Introduction

- Ask students to name as many simple shapes as possible, focus on shapes with equal sides and angles.
- For each shape:
  - How would you explain to someone how to draw that shape?
  - How could you draw this using a loop?

ACTIVITY

Artist: Loops

In the Artist levels students will no longer be constrained to 90 degree angles. Having protractors available can be help students better visualize the angles they need.

EXTENDED LEARNING

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

The Copy Machine

- Give students two pieces of paper.
- On one sheet have the students draw a shape with equal sides and angles.
- On the second sheet draw instructions for recreating that shape using loops.
- Trade instruction sheets and attempt to recreate the shape using only the provided instructions.
  - Can you predict what shape will be drawn just by reading the instructions?

8

Course 2 | Lesson 8

Bee: Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

In the last stage students used loops to repeat simple movements. Now they're going to add to that the looping of actions in order to help the bee collect more nectar and make more honey.

LESSON OBJECTIVES

Students will:

- Write a program for a given task which loops a single command
- Identify when a loop can be used to simplify a repetitive action
- Employ a combination of sequential and looped commands to move and perform actions

GETTING STARTED

Introduction

At this point, students have used loops in context of both the Maze and the Artist levels. Both of those stages focused on looping movement instructions.

- What are some other elements of our programs that could benefit from loops?
- How do you think we could use loops to make the Bee programs more efficient?

ACTIVITY

Bee: Loops

When students are using loops to repeat an action (such as getting nectar), encourage them to think about the movements before and after that action. Could those be brought into the loop as well?

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!

U

UNPLUGGED

Relay Programming

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON 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 create a program, one instruction at a time.

LESSON OBJECTIVES

Students will:

- Practice communicating ideas through codes and symbols
- Use teamwork to complete a task
- Verify the work of their teammates to ensure a successful outcome

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Blank Paper or Index Cards for programs
- Sheets of 4x4 paper grids for the students to use as practice (These are provided as part of the Relay Programming Activity Packet, but if you have the students create their own, you can include Common Core Math standard 2.G.2.)
- Markers, pens, or pencils

- Relay Programming Assessment

For the Teacher

- Lesson Video
- Locate a wide open space for this activity, such as the gym or outdoor field
- Print out one Relay Programming Activity Packet for each group
- Print one Relay Programming Assessment for each student
- Supply each group with plenty of paper and pens/pencils

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. You can do this as one large group or have students discuss with an elbow partner.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) Vocabulary

This lesson has one new and important word:

Debugging - Say it with me: De-bugg-ing
Finding and fixing problems in your algorithm or program

3) Refresh Graph Paper Programming

Recall that in "Graph Paper Programming" we guided our teammates' Automatic Realization Machine (ARM) using arrows. Take a moment to do a refresher Graph Paper Programming image -- either one that you have already covered or an entirely new one.

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.

ACTIVITY: RELAY PROGRAMMING (20 MIN)

4) Relay Programming

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:

1. Divide students into groups of 3-5.
2. Have each group queue up relay-style.
3. Place an identical image at the other side of the room/gym/field from each team.
4. 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.
5. The first student then runs back and tags the next person in line, then goes to the back of the queue.
6. 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.
7. That student then dashes back to tag the next person, and the process continues until one group has finished their program.
8. First group to finish is the winner!

Play through this several times, with images of increasing difficulty.

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, they must also cross out the rest of the program after that. This counts as their entire turn. The next player is allowed to continue from the last *correct* instruction.

WRAP-UP (10 MIN)

5) 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?

6) Vocab Shmocab

You can choose to do these as a class, or have the students discuss with an elbow partner and share.

- Which one of these definitions did we learn a word for today?

  "To rub or scrape out letters or characters"
  "Doing something repeatedly in a similar way"
  "Finding and fixing problems in your algorithm or program"

  ...and what is the word that we learned?

- Which of these things could you debug?

  *The wrong answer to a math problem
  *A tunafish sandwich
  *Two baskets of pine cones

  Explain why you chose your answer.

ASSESSMENT (10 MIN)

7) Relay Programming Assessment: Debugging

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.

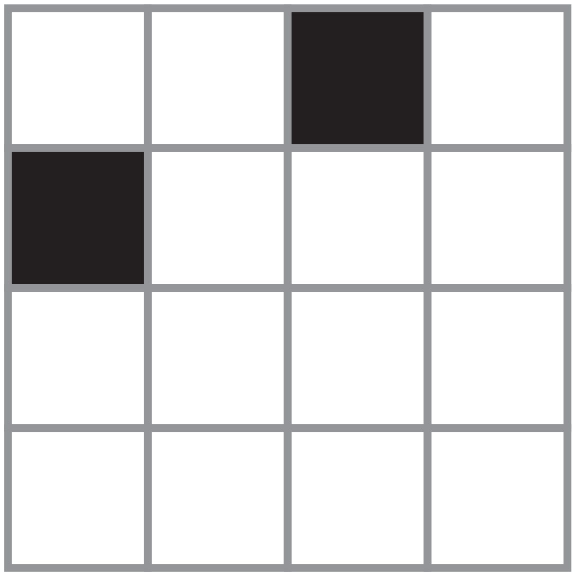
Debugging Together

- As the teacher, 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.
    - How many students had the same program?
    - Anyone have something different?

1

Relay Programming

Relay Image 1

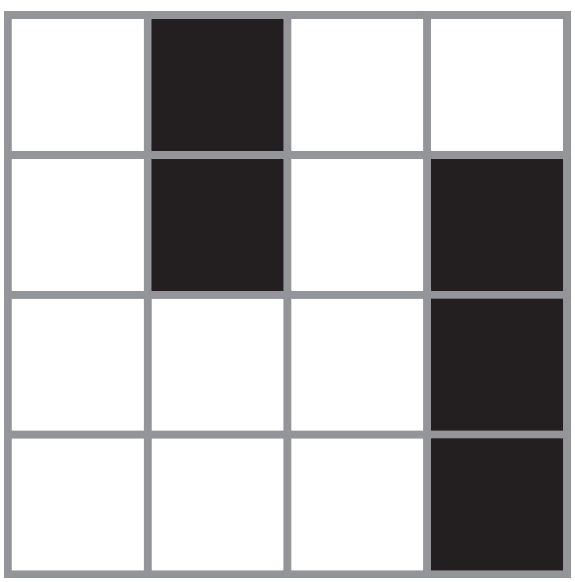


Revision 140710.1a

2

Relay Programming

Relay Image 2

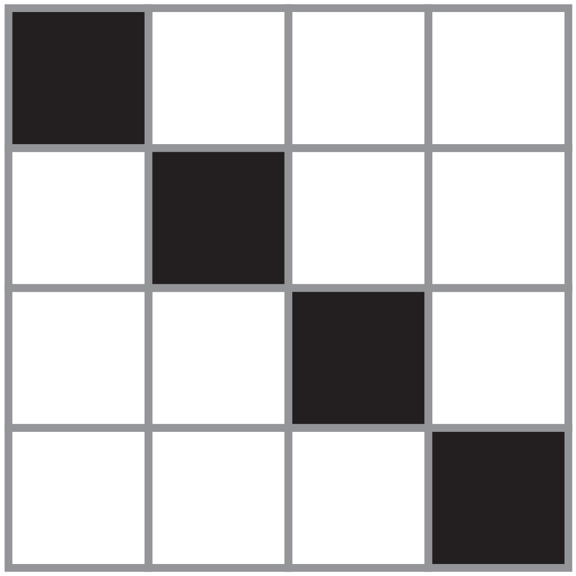


Revision 140710.1a

3

Relay Programming

Relay Image 3

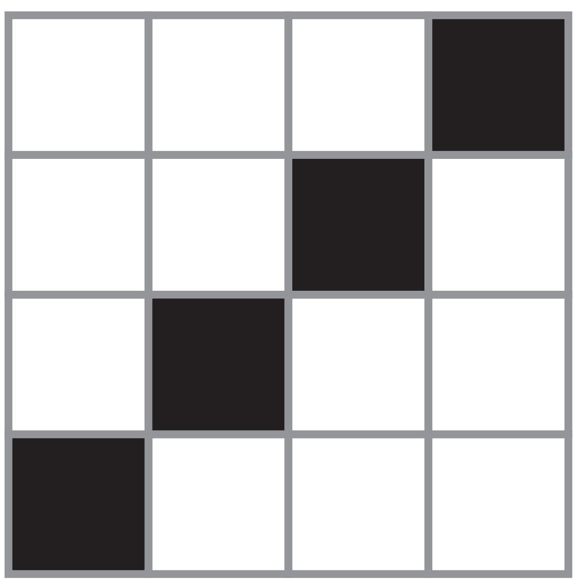


Revision 140710.1a

4

Relay Programming

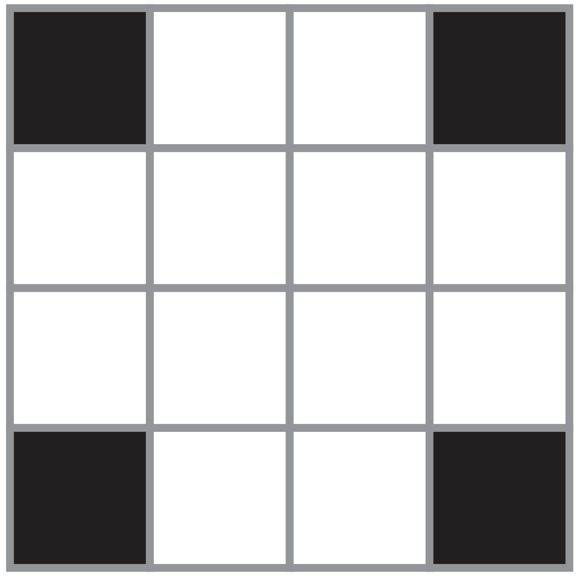
Relay Image 4



Revision 140710.1a

5 Relay Programming

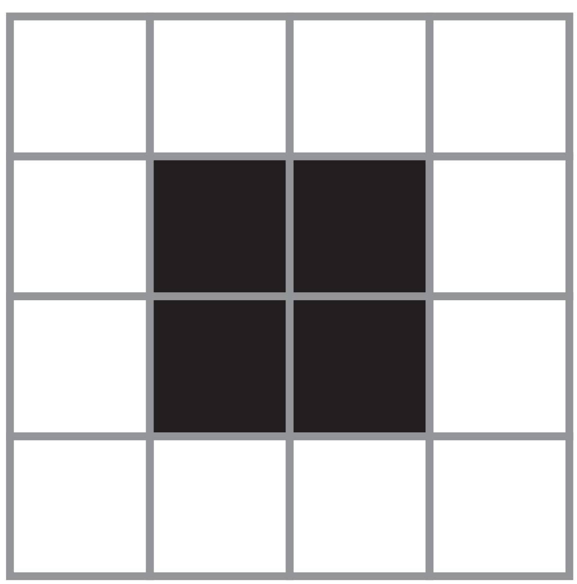
Relay Image 5



Revision 140710.1a

6 Relay Programming

Relay Image 6



Revision 140710.1a

Name: ____________________ Date: ______________

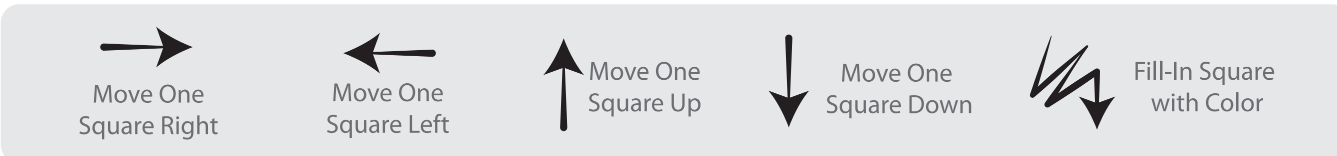
Debugging

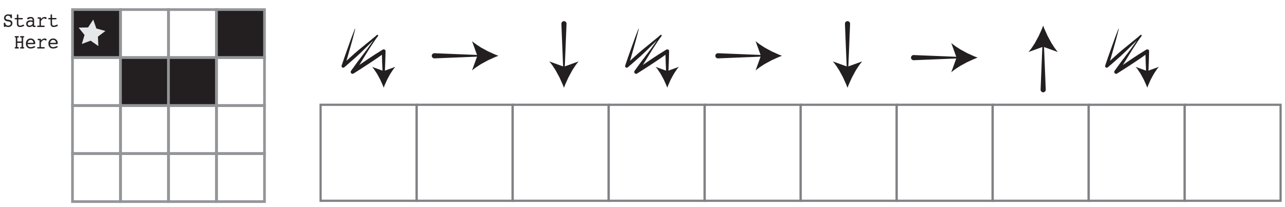
Assessment Worksheet

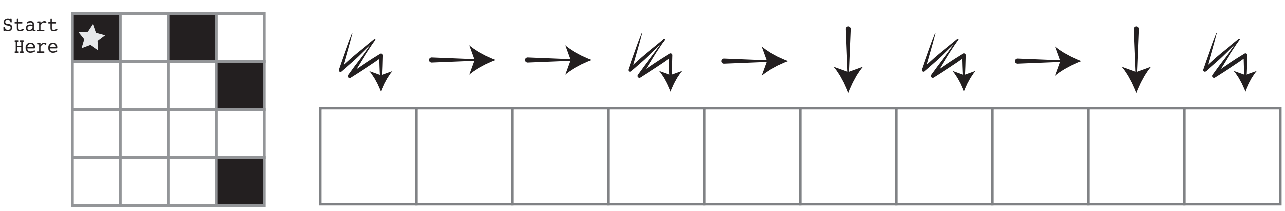
Sometimes when you are coding in groups, someone will make an error that will affect everyone.

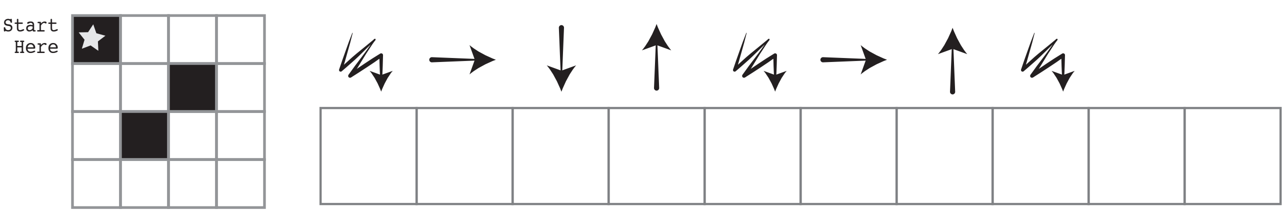
Somebody has already written programs for the images below, but each one has a mistake! Figure out what the programs are *supposed* to look like, and circle the error in each one. Then, draw the correct symbol in the box beneath.

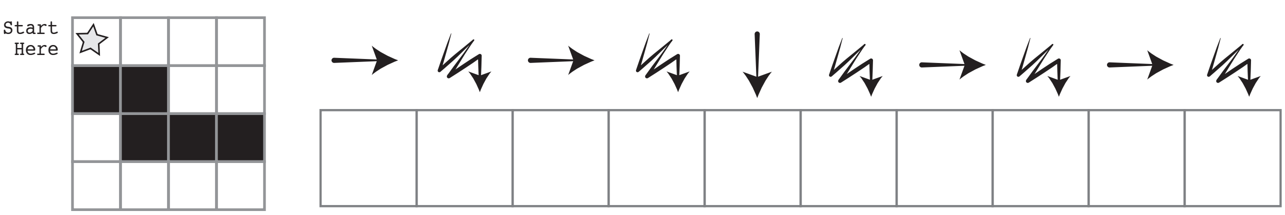
Each program should use the symbols below to draw the image to its left.











10

Course 2 | Lesson 10

Bee: Debugging

Lesson time: 30 Minutes

LESSON OVERVIEW

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 misordered blocks.

LESSON OBJECTIVES

Students will:

- Predict where a program will fail
- Modify an existing program to solve errors
- Identify an algorithm that is unsuccessful when the steps are out of order
- Reflect on the debugging process in an age-appropriate way

GETTING STARTED

Introduction

Ask students to think about problems they have to solve in everyday life.

- How do you fix something that isn't working?
- Do you follow a specific series of steps?
- The puzzles in this unit have already been solved for you (yay!), but they don't seem to be working (boo!)
- We call the problems in these programs "bugs," and it will be your job to "debug" them.

ACTIVITY

Bee: Debugging

As your students work through the puzzles, observe how they search for bugs. Identify different strategies used and ask students to share with the whole class. This helps students to recognize that there are many ways to approach these problems. Have students follow the path described by the code with their fingers to find potential bugs.

EXTENDED LEARNING

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

Planting bugs

Have students go back through previous levels, purposefully adding bugs to their solutions. They can then ask other students to debug their work. This can also be done with paper puzzles.

11

Course 2 | Lesson 11

Artist: Debugging

Lesson time: 30 Minutes

LESSON OVERVIEW

In this stage, students will continue practicing their debugging skills by helping the Artist to fix pictures that aren't coming out quite right.

LESSON OBJECTIVES

Students will:

- Predict where a program will fail
- Modify an existing program to solve errors
- Identify an algorithm that is unsuccessful when the steps are out of order
- Reflect on the debugging process in an age-appropriate way

GETTING STARTED

Introduction

By now students should be pretty comfortable digging in and finding bugs. This is a great time to bring the class together to share debugging tactics and difficulties.

- What kinds of bugs are easiest for you to see? Why?
- Which bugs were the hardest to find? How did you eventually fix them?
- What's the first thing you look for in a buggy program?

ACTIVITY

Artist: Debugging

Some students are averse to running a program until they've fixed it. Sometimes the easiest way to figure out what's wrong with a program is to watch it fail, so there's nothing wrong with running a program before we've finished fixing it. The only time we care if they got it right the first time is on the assessment levels.

EXTENDED LEARNING

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

Planting bugs

Have students go back through previous levels, purposefully adding bugs to their solutions. They can then ask other students to debug their work. This can also be done with paper puzzles.

U

UNPLUGGED

Conditionals with Cards

Lesson time: 30 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

We don't always know ahead of time what things will be like when we run our computer programs. Different users have different needs, and sometimes you will want to do something based off of one user's need that you don't want to do with someone else. That is where conditionals come in. This lesson demonstrates how conditionals can be used to tailor a program to specific information.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review 2) Vocabulary
3) On One Condition

Activity: Conditionals with Cards - 30 minutes

4) Conditionals with Cards

Wrap-up - 10 minutes

5) Flash Chat - What did we learn? 6) Vocab Shmocab

Assessment - 5 minutes

7) Conditionals with Cards Assessment

LESSON OBJECTIVES

Students will:

- Define circumstances when certain parts of programs 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

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Playing Cards
- Paper for keeping track of how a program reacts to a card
- Pens & Pencils
- Conditionals with Cards Assessment

For the Teacher

- Lesson Video
- This Teacher Lesson Guide
- One Sample Program for the class to look at
- Print one Conditionals with Cards Assessment for each student

GETTING STARTED (20 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. You can do this as one large group or have students discuss with an elbow partner.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP *Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.*

2) 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

3) On One Condition

- 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

        - or Do a handstand
    - Start counting right away.
    - If the students succeed, point out right away that they succeeded, so they *do* 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 7, 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.

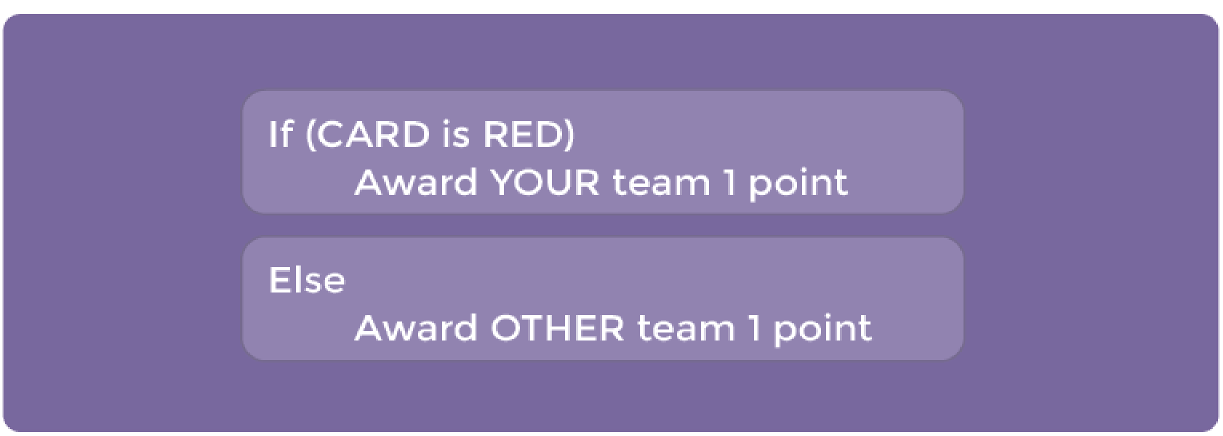
ACTIVITIES: (20 MIN)

4) [Conditionals with Cards]

Directions:

1) 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:



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;
}
```

2) Decide how you want to split your class into teams.

3) Each team should have a pile of cards (at least as many cards as team members) nearby.

4) Put one of your “Programs” up on the board for all to see.

5) Have the teams take turns drawing cards and following the program to see how many points they score in each round.

6) 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 (5 MIN)

5) 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?
- 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.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

"Adding additional space to the beginning of a line of text"
"A combination of yellow and green"
"Statements that only run under certain conditions"

...and what is the word that we learned?

ASSESSMENT (5 MIN)

7) 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.

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:
  - If the suit is clubs
  - If 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;
}
```

U

Conditionals with Cards

Sample Program 2

Sample program as algorithm

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

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;
        }
}
```

Name: ______________________ Date: ______________

U

Unplugged

Conditionals with Cards

Assessment Activity

CODE

Look at the program below.

The steps below show each team taking turns to play the Conditionals Game. See if you can figure out what happens for each draw. Write down the score during each round along the way. After three rounds, circle the winner.

```
If (CARD is lower than 5)
        If ( CARD is BLACK)
                Award YOUR team the same
                number of points on the card.

        Else
                Award OTHER team 1 point.

Else
        If ( CARD is HEARTS)
                Award YOUR team 1 point
```

Here's how the game went:

	TEAM #1	END OF ROUND SCORE	TEAM #2	END OF ROUND SCORE
		0		0
ROUND #1	3 ♠	___	7 ♥	___
ROUND #2	4 ♥	___	4 ♣	___
ROUND #3	9 ♣	___	5 ♦	___

13

Course 2 | Lesson 13

Bee: Conditionals

Lesson time: 30 Minutes

LESSON OVERVIEW

Up until this point all the programs your students have written should run exactly the same way every time - reliable, but not very flexible. In this stage we introduce the conditional statement, code that functions differently depending on the conditions it encounters.

LESSON OBJECTIVES

Students will:

- Compare values using the = operator
- Translate spoken language conditional statements into a program
- Identify when a conditional can be used to deal with unknown values
- Execute an algorithm with a conditional statement
- Solve puzzles using a combination of looped sequences and conditionals

GETTING STARTED

Introduction

Review the Conditionals with Cards activity with your students.

- What is a conditional statement?
- When is it useful?
- What are some of the conditions you used in the Unplugged activity?

Now we're going to use conditionals with our bee to help us deal with some mysterious purple flowers. We don't know if those flowers have nectar or not, so we'll need to use conditionals to make sure that we collect nectar if it's there, but that we don't try to collect nectar from a flower that doesn't have any.

ACTIVITY

Bee: Conditionals

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:
  - If the suit is clubs
  - If 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?

U

UNPLUGGED

Binary Bracelets

Lesson time: 15 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

Binary is extremely important to the computer world. The majority of computers today store all sorts of information in binary form. This lesson helps to demonstrate how it is possible to take something that we know and translate it into a series of ons and offs.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) Off and On

Activity: Binary Bracelets - 15 minutes

4) Binary Bracelets

Wrap-up - 5 minutes

5) Flash Chat - What did we learn?

Assessment - 10 minutes

6) Binary Assessments

LESSON OBJECTIVES

Students will:

- Encode letters into binary
- Decode binary back to letters
- Relate the idea of storing initials on a bracelet to the idea of storing information in a computer

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Binary Bracelet Worksheet
- Binary Assessment
- Pens and Pencils
- Scissors

For the Teacher

- Lesson Video
- This Teacher Lesson Guide
- Binary Bracelet Worksheet
- Binary Assessment
- Computer for opening or images of an open computer
- Optional: Write a short message on the board in binary

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

Here are some questions that you can ask in review:

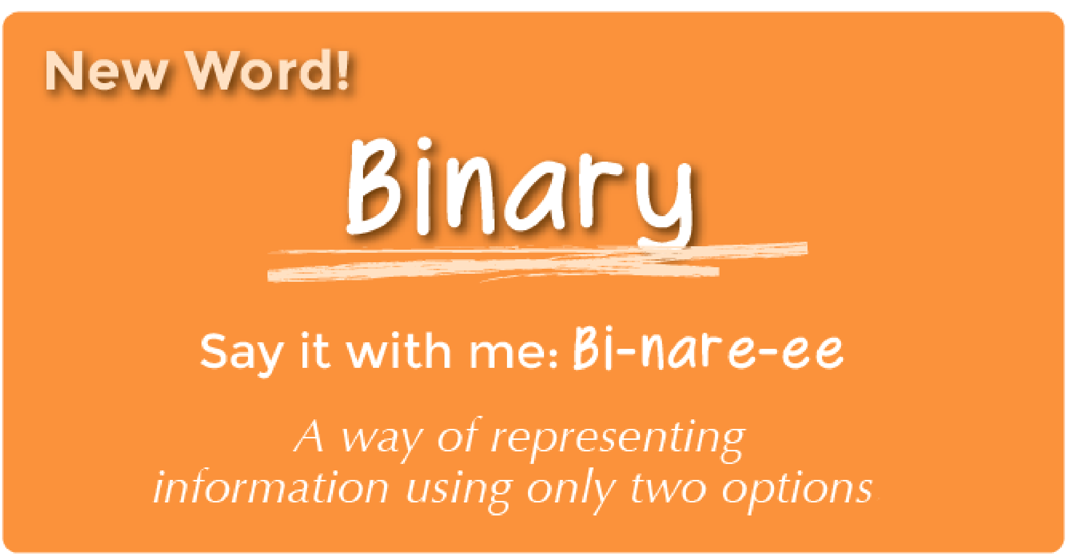
- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) Vocabulary

This lesson has one new and important word:



Binary - Say it with me: Bi-nare-ee
A way of representing information using only two options

3) 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."
        - 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 the things we store in a computer into binary?
        - Let's start with letters.
        - Use the Binary Decoder Key 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.**
        - 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 your encoded letters on their own, you can move on to the activity.

ACTIVITY: (20 MIN)

4) Binary Bracelet Worksheet

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.*

*You **do not** need to cover the whole of binary, including counting and converting numbers back and forth from decimal. i n is intended to be a fun introduction to how computers store information, not a frustrating lesson in bases.*

Directions:

1. Find the first letter of your first name in the Binary Decoder Key.
2. Fill in the squares of the provided bracelet to match the pattern of the squares next to the letter that you selected.
3. Cut the bracelet out.
4. Tape the bracelet around your wrist to wear it!
5. Share your bracelet with your classmates to see if they can figure out your letter.

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	■□■□	□■□■

LESSON TIP

ing this exiercise using thread i ners an 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.

After the activity, revisit the message that was on the board and see if your class can decypher it using what they've learned.

WRAP-UP (5 MIN)

5) 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?

ASSESSMENT (15 MIN)

7) Binary 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.

Name: ____________________ Date: ______________

U

Unplugged

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!

□□□□ □□□□

Name: ____________________ Date: ______________

Unplugged

Binary Bracelets

Assessment for Binary Bracelets Lesson

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?

■□■■ ■■□□ ______

■□■■ □□□□ ______

■□■■ ■□■■ ______

■□■■ ■□■□ ______

■□■■ □■■□ ______

■□■□ ■■□□ ______

■□■■ ■□□■ ______

■□■□ ■□■□ ______

■□■■ □□□■ ______

______ ______ ______ ______

______ ______

______ ______ ______

U

UNPLUGGED

The Big Event

Lesson time: 15 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON 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.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) A Series of Events

Activity: Events - 15 minutes

4) The Big Event

Wrap-up - 5 minutes

5) Flash Chat - What did we learn?

Assessment - 10 minutes

6) The Big Event Assessment

LESSON OBJECTIVES

Students will:

- 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

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Assessment Worksheet: The Big Event Assessment
- Pens/Pencils/Markers

For the Teacher

- Lesson Video
- Teacher Lesson Guide

- Print one The Big Event Activity Worksheet and Event Controller
- Print Assessment Worksheet: The Big Event Assessment for each student

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) 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.

3) 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.

ACTIVITY: (15 MIN)

4) The Big Event

- Do you remember guiding your friends to fill in an image of squares in Graph Paper Programming?
  - In that exercise, you knew in advance exactly what you wanted your friends to draw, 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.

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!

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.

Directions:

1. Project the Event Controller onto your classroom screen.

1. 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
2. Practice tapping the buttons on the overhead and having your class react.
3. Add some button sequences into the mix and have the students try to keep up with their sounds.
4. 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.
5. Get the class started on a planned task before interrupting them again with the buttons. We suggest:
   - Counting to 10
   - Singing "Old MacDonald"
6. Once their plan is underway, interject button presses sporadically.
7. 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)

5) 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)

7) Assessment Worksheet: Controlling by Events 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

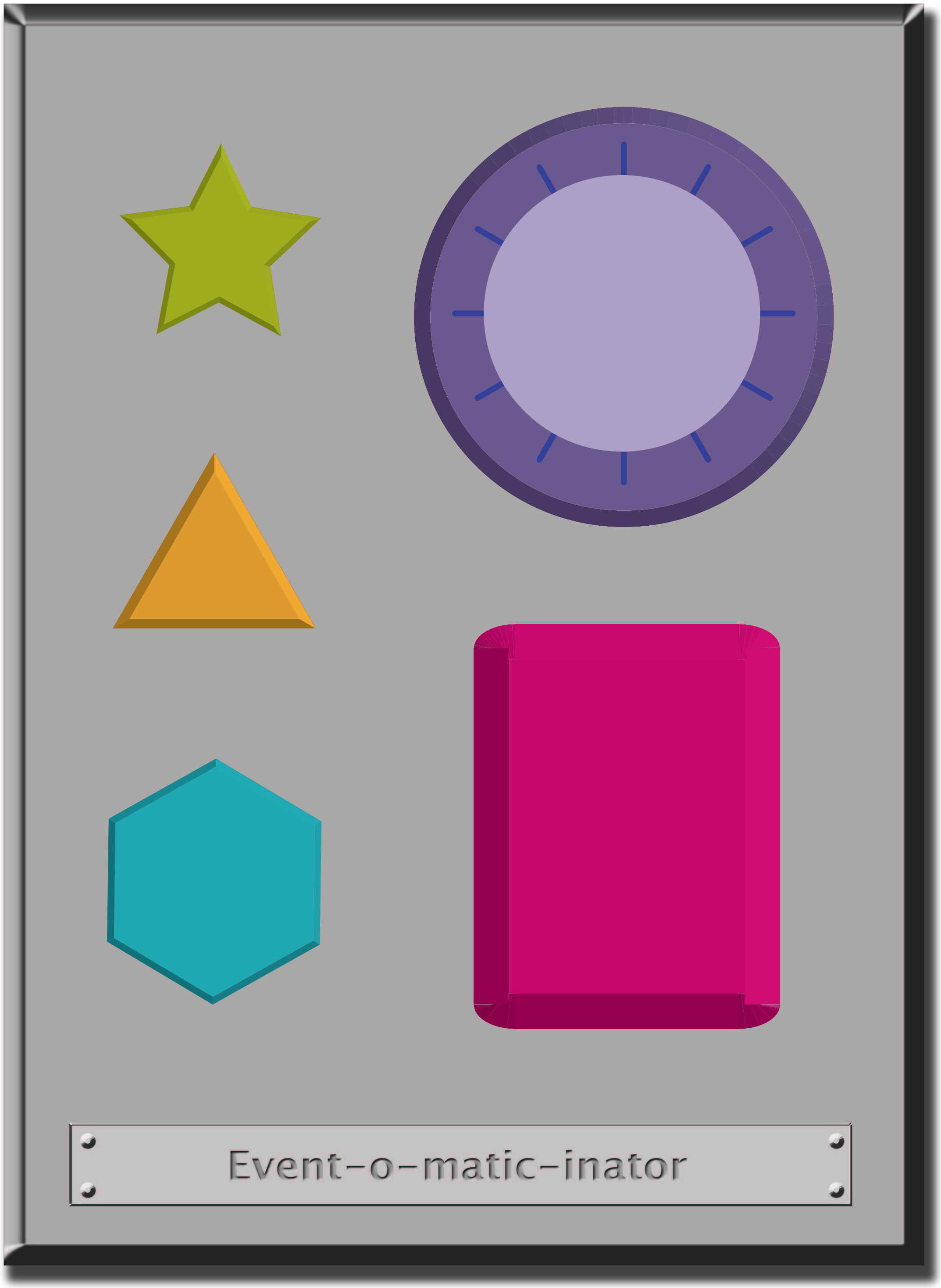
- Assign each student an event to watch out for, and an appropriate reaction to that event. Chain the actions so that each child's reaction becomes an event that triggers the reaction of another student. Keep assigning until everyone has something to do and everyone makes someone react.

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!

The Big Event

Event Controller



Name: ______________________ Date: ______________

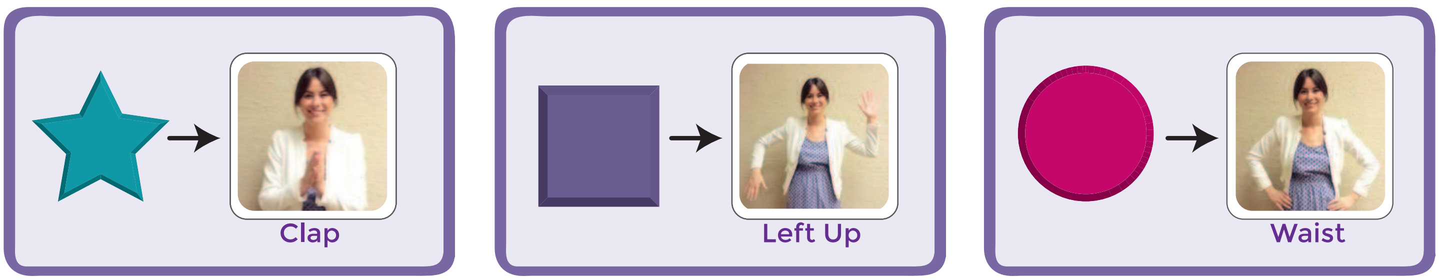
Unplugged

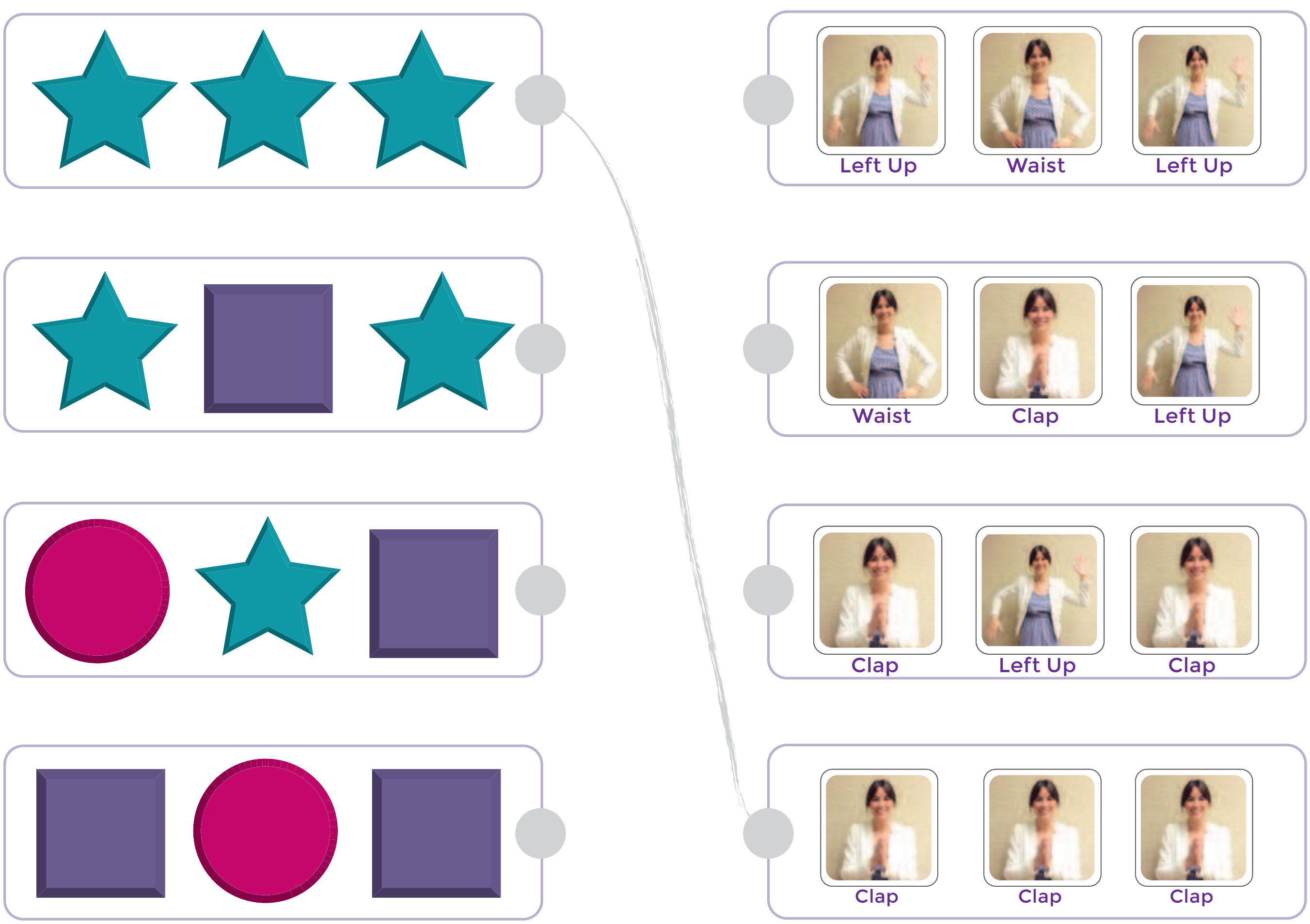
The Big Event

Controlling by Events Assessment

You've been given a magical controller that makes your principal do funny things with her arms.

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





16

Course 2 | Lesson 16

Flappy

Lesson time: 30 Minutes

LESSON OVERVIEW

In this special level students get to build their own Flappy game by using event handlers to detect mouse clicks and object collisions.

LESSON OBJECTIVES

Students will:

- Match blocks with the appropriate event handler
- Create a game using event handlers
- Share a creative artifact with other students

GETTING STARTED

Introduction

- Review The Big Event activity with students:
  - What did we "program" the button click events to do?
- Now we're going to add events to our coding. Specifically, we're going to create an event for clicking the mouse and the bird hitting an object.
  - In video game programming we call this kind of event collision detection; it lets us decide what to do when one thing collides with, or touches, another.
  - What kinds of collision events have you seen in games?

LESSON TIP

Students will have the opportunity to share their final product with a link. This is a great opportunity to show your school community the great things your students are doing. Collect all of the links and keep them on your class website for all to see!

ACTIVITY

Flappy

In the final stage of this lesson students are able to tweak their game to make it unique - encourage them to see how different they can make each game within the constraints provided.

EXTENDED LEARNING

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

Look Under the Hood

When you share a link to your game, you also share all of the code that goes behind it. This is a great way for students to learn from each other.

- Post links to completed games online or on the board.
  - Make a game of your own to share as well!
- When students load up a link, have them click the "How it Works" button to see the code behind the game.
- Discuss as a group the different ways your classmates coded their games.
  - What suprised you?
  - What would you like to try?
- Choose someone else's game and build on it. (Don't worry; the original game will be safe.)

17

Course 2 | Lesson 17

Play Lab: Create a Story

Lesson time: 30 Minutes

LESSON OVERVIEW

In this culminating plugged activity, students will have the opportunity to apply all of the coding skills they've learned to create an animated story. It's time to get creative and create a story in the Play Lab!

LESSON OBJECTIVES

Students will:

- Identify actions that correlate to input events
- Create an animated, interactive story using sequence, loops, and event-handlers
- Share a creative artifact with other students

GETTING STARTED

Introduction

Review the event handling students did in Flappy:

- What did events did you use in coding Flappy?
- Now you're going to animate multiple characters using events triggered by the arrow keys to tell a story.
- This is your chance to get really creative!

LESSON TIP

Students will have the opportunity to share their final product with a link. This is a great opportunity to show your school community the great things your students are doing. Collect all of the links and keep them on your class website for all to see!

ACTIVITY

Play Lab: Create a Story

This is the most free-form plugged activity of the course. At the final stage students have the freedom to create a story of their own. You may want to provide structured guidelines around what kind of story to write, partiularly for students who are overwhelmed by too many options.

EXTENDED LEARNING

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

Look Under the Hood

When you share a link to your story, you also share all of the code that goes behind it. This is a great way for students to learn from each other.

- Post links to completed stories online or on the board.
  - Make a story of your own to share as well!
- When students load up a link, have them click the "How it Works" button to see the code behind the story.
- Discuss as a group the different ways your classmates coded their stories.
  - What suprised you?
  - What would you like to try?
- Choose someone else's story and build on it. (Don't worry; the original story will be safe.)

U

UNPLUGGED

Your Digital Footprint

Lesson time: 30 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

In collaboration with Common Sense Media, 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.

TEACHING SUMMARY

Getting Started - 20 minutes

1) Review 2) Vocabulary
3) Pause and Think

Activity: Follow the Digital Trail - 30 minutes

4) Follow the Digital Trail

Wrap-up - 10 minutes

5) Flash Chat - What did we learn? 6) Vocab Shmocab

Assessment - 5 minutes

7) Digital Footprint Assessment

LESSON OBJECTIVES

Students will:

- Understand that being safe when they visit websites is similar to staying safe in real life
- Learn to recognize websites that are alright 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

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- One Animal Tracks Chart (page 7)
- Pens & Pencils
- Digital Footprint Assessment

For the Teacher

- Lesson Video
- This Teacher Lesson Guide
- Download or prepare the "Pause and Think" video
- Common Sense Media's Follow the Digital Trail game
- Print one set of Animal Tracks from this PDF
- Print one Animal Tracks chart (page 7) for each student
- Print one Digital Footprint Assessment for each student

GETTING STARTED (20 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. You can do this as one large group or have students discuss with an elbow partner.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) Vocabulary

This lesson has one new and important word:

Digital Footprint - Say it with me: Dih-jih-tal Foot-print
The information about someone on the Internet

3) 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 Pause and Think Online 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 ng extreme caution with your n i

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

ACTIVITIES: (20 MIN)

4) Follow the Digital Trail

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

	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 user name 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?		

Directions:

1) 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.

2) 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.

3) Explain to students that they are from the "Things Big and Small" Detective Agency. An evil 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.

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

5) 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

For more in depth modules, you can find additions to this curriculum at the ommon ense edia page on cope and e uence.

WRAP-UP (5 MIN)

5) 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:

Appropriate	Inappropriate
Interests	Address
Hobbies	Full name
First name	Information that would hurt others

LESSON TIP

Flash hat uestions are intended to spark big pictthinking about how the lesson relates to the greater world and the students' greater future. se your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

"The information about someone on the Internet"
"Changing the sun's light to energy"
"A person who is under the age of 21"

...and what is the word that we learned?

ASSESSMENT (5 MIN)

7) 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 Media

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

Using Materials From common sense

Name: ______________________ Date: ______________

Unplugged

Your Digital Footprint

Staying Safe and Responsible Assessment

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.

	A) Fran the Fish	B) Betty the Bird	C) Tony the Tiger
Hobbies	swimming	flying	going to the 3rd Street gym
Address	the sea	a nest	523 Green Street
Other	pet's name is Frank	I love seeds!	My real name is Thomas / Hello, I'm Thomas

Circle One:

A) Fran the Fish

B) Betty the Bird

C) Tony the Tiger

19

Course 2 | Lesson 19

Artist: Nested Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Students use the Artist environment to write programs that have looped statements inside another loop, which is called a nested loop.

LESSON OBJECTIVES

Students will:

- Count the number of times an action should be repeated and represent it as a loop
- Divide the number of degrees in a circle into even segments
- Given a number of segments, calculate the degrees need to complete a circle
- Break complex tasks into smaller repeatable sections
- Combine simple shapes into complex designs with nested loops

GETTING STARTED

Introduction

ACTIVITY

Artist: Nested Loops

Course 3

OVERVIEW

Students create programs with different kinds of loops, events, functions, and conditions and write algorithms for everyday tasks. Through this they will investigate different problem-solving techniques, discuss societal impacts of computing and the Internet, and learn about Internet transmission methods. By the end of the curriculum, students create interactive stories and games they can share with anyone. Students taking Course 3 will have already taken Course 2.

Lesson Sequence of Course 3

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

#	Lesson Name	Description
1	**Computational Thinking**	Students use the steps of computational thinking (decompose, pattern match, abstract, algorithm) to figure out how to play a game that comes with no instructions.
2	Maze	Students write programs (an algorithm for the computer) that get a character through a maze. They'll understand the importance of sequence and basic loops (repeated statements) in the programs they write.
3	Artist	Students write programs to draw different shapes.
4	**Functional Suncatchers**	Students create an algorithm with functions (pieces of code that you want to use over and over again) to create suncatchers using string and beads.
5	Artist: Functions	Using and modifying prebuilt procedures in the the Artist environment, students gain familiarity with how code is written for functions.
6	Bee: Functions	Using the Bee environment, students use and modify functions to help the bee collect nectar and make honey.
7	Bee: Conditionals	In the Bee environment, students write programs with conditional statements. Students originally learned this concept in Course 2, but this lesson introduces more complex implementations of conditionals.

#	Lesson Name	Description
8	Maze: Conditionals	Using the Maze environment, students write programs using conditionals.
9	**Songwriting**	Students use the concept of the chorus in a song to learn about functions.
10	**Real-Life Algorithms - Dice Race**	This lesson calls out ways we use algorithms in our daily lives. Students have to identify and write down the algorithm for a dice race game.
11	Artist: Nested Loops	Students use the Artist environment to write programs that have looped statements inside another loop, which is called a nested loop.
12	Farmer: While Loops	Using while loops, students control a farmer shovel dirt into holes until they're full and remove dirt from piles until it's all gone.
13	Bee: Nested Loops	Students use the Bee environment to write programs using nested loops.
14	Bee: Debugging	Using the same environment as the prior online activity, students are presented with a pre-written program that fails to complete the puzzle. Students will have to "debug" or fix the pre-written program.
15	**Bounce**	Using the concept of "Events," (a concept learned in Course 2) students will create a game of their own with events like "When the ball goes through the goal, you score a point."
16	Play Lab: Create a Story	Students use the Mini-Studio environment to create their own interactive stories.
17	Play Lab: Create a Game	Students use the Mini-Studio environment to create their own interactive games.
18	**Internet**	Students send messages representing various Internet transmission methods using pieces of paper.
19	**Crowdsourcing**	Student use crowdsourcing, a problem-solving technique common in computer science, to complete a task together as a classroom more efficiently than if a student attempted it alone.
20	**Digital Citizenship**	Students explore the difference between private information and personal information, distinguishing what is safe and unsafe to share online.

#	Lesson Name	Description
21	Artist: Patterns	Students write programs that draw interesting and beautiful patterns.

U

UNPLUGGED

Computational Thinking

Lesson time: 25 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

For this activity, no instructions are provided. Instead, students will use examples of what imaginary players have done to figure out how to play the game. This lesson gives students the opportunity to practice the four arts of computational thinking (decomposition, pattern matching, abstraction, and algorithms) in one cohesive activity.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Vocabulary
2) Figuring it Out

Activity: Computational Thinking - 25 minutes

3) Computational Thinking

Wrap-up - 10 minutes

4) Flash Chat - What did we learn?
5) Vocab Shmocab

Assessment - 5 minutes

6) Computational Thinking Assessment

LESSON OBJECTIVES

Students will:

- Analyze information to draw conclusions
- Match identical portions of similar phrases to match patterns
- Identify differences in similar phrases and abstract them out

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- One die per group
- One Computational Thinking Kit per group
- Pens, Pencils, & Scissors
- Computational Thinking Assessment for each student

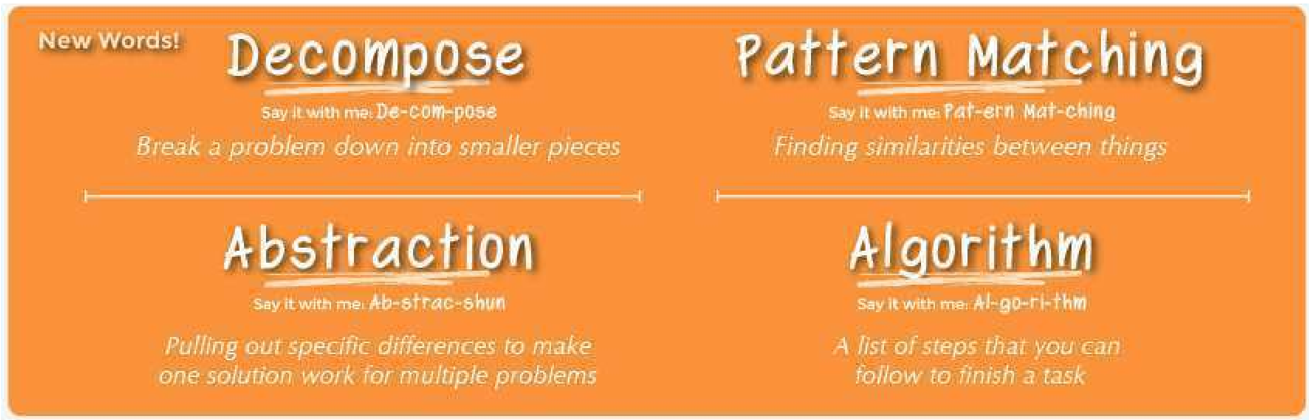
For the Teacher

- Lesson Video
- This Teacher Lesson Guide
- Print one Computational Thinking Kit per group
- Print one Computational Thinking Assessment for each student

GETTING STARTED (15 MIN)

1) Vocabulary

This lesson has four 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

Decompose - Say it with me: De-com-pose
Break a problem down into smaller pieces

Abstraction - Say it with me: Ab-strac-shun
Pulling out specific differences to make one solution work for multiple problems

Pattern Matching - Say it with me: Pat-ern Matching
Finding similarities between things

2) Figuring it Out

- Tell your students that you want them to sum up all of the numbers between 1 & 200.
  - Use your body language to indicate that this is not a "serious" or graded exercise.
    - Now, let them know that they must do it all in their heads.
    - Add the time constraint of thirty seconds.
    - They may feel overwhelmed. This is intentional. You can indicate with your tone and demeanor that you might be crazy asking this of them, but begin timing with a resounding: "Starting NOW".
- Watch the class as you keep time. How many are lost in thought?
- When time is up, ask if anyone was able to get the total.
- Ask if there is anyone who thought the problem was so hard that they didn't even attempt it.
- Did anyone attempt it and just not finish?
  - What did they try?
- Guide students toward thinking a little smaller.
  - If we break the problem up into smaller pieces, it becomes easier to manage.
  - Let's start at the two ends. What is 200 + 1?
  - What is 199 + 2?
  - What is 198 + 3?
  - See a pattern?
  - How many of these pairs will we have?
    - What is the last pair we will find? 100 + 101

        - That means that we have 100 total pairs.
        - If we have 100 total pairs of sums of 201, how do we find the final total?
        - What is 100 * 201?
            - Now, what if we wanted to find the trick to do this with other numbers?
    - Can we do it easily with 2,000?
    - How about 20,000?
    - What stays the same? What is different?
    - If we use abstractions to make our end goal something that can change (say we name it "blank") then we can make an algorithm that will work for any number
- Work through the problem until you ultimately get ? = ("blank"/2) * ("blank"+1)
- Do a few simple examples to show that the algorithm is correct for blanks= 2, 3, 4, & 5.

"This is all to show that if you use the tools of Computational Thinking (decomposition, pattern matching, abstraction, and algorithms), then you can figure out how to solve problems that no one has already taught you how to solve...just like we did here! This will be an extremely powerful skill for the rest of your life!"

ACTIVITIES: (25 MIN)

3) Computational Thinking

This lesson is all about a "Game with No Instructions." Students will be charged with figuring out how to play the game as a small group. The small details of their final algorithm are unimportant. What *is* important is that they were able to take a huge task like "figuring out how to play a game on their own" and take small steps toward achieving the goal.

Students will be guided toward discovering the rules using the steps of computational thinking. Resist the temptation to point the students toward "doing it right" and allow them just to do it on their own. If they feel stumped or confused, encourage the students to look at the information that has been given to them, or if they must, ask a classmate.

Directions:

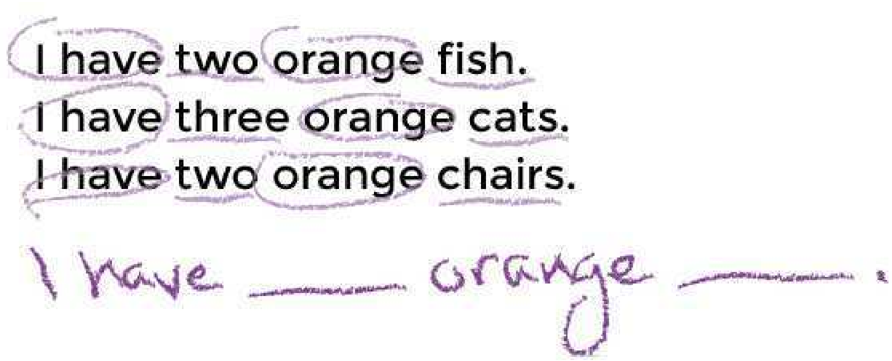
1) Divide students into groups of 2-4.

2) Have the groups read over user experiences to get an idea of how other students have played the “Game with No Instructions.”

3) Encourage them to pattern match between each experience by circling the sections of words that are identical from player to player.

4) Next, have them abstract away differences from each experience by underlining words that change from player to player.

5) Using pattern matching and abstraction, have them make a script template for game play by writing up the circled parts of the other students’ experiences, and leaving the underlined sections as blanks.



For example:

6) Give students a blank sheet of paper to write a list of instructions for how they think this game should be played based on the user experiences that they just read. This will be their algorithm.

7) Have students play the game using the algorithm that they just made. Each player should get at least two turns.

WRAP-UP (5 MIN)

4) Flash Chat: What did we learn?

- What should you try to do when you're asked to do something and you don't know how?
- If a problem is too hard, what should you try to do?
- If you find similarities in lots of solutions to different problems, what does that probably tell you?
- If you have a problem that is just a little different from a problem that you have a solution for, what would you 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.

5) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

"Bringing two pieces together"
"Breaking a problem down into smaller pieces"
"An educated guess"

...and what is the word that we learned?

ASSESSMENT (5 MIN)

6) Computational Thinking 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.

Created in partnership with THINKERSMITH®

Name: ____________________ Date: ____________

U

Unplugged

Computational Thinking

User Experience Scripts

Figure out how to play this game by looking at the players' phrases below. Circle the matching parts and underline words that are different from player to player. The first matching section has been circled for you.

Player 1:

"I chose a lion, and rolled a six, then a four, then a two. That means I need to draw a black cupcake on my lion's tail."

Player 2:

"I chose a donkey, and rolled a three, then a two, then a one. That means I need to draw a yellow pineapple on my donkey's head."

Player 3:

"I chose a puppy, and rolled a five, then a three, then a five. That means I need to draw a pink salmon on my puppy's nose."

Using pattern matching and abstraction, make yourself a template for game play by writing up the circled parts of the other students' experiences, and leaving the underlined sections as blanks.

"I chose a ____________________

____

____

____

____

Computational Thinking

Roll Catalog

Color:

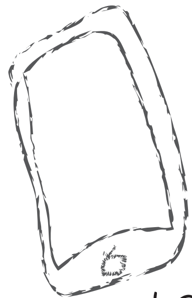
1) Red
2) Blue
3) Yellow
4) Green
5) Pink
6) Black

Items:

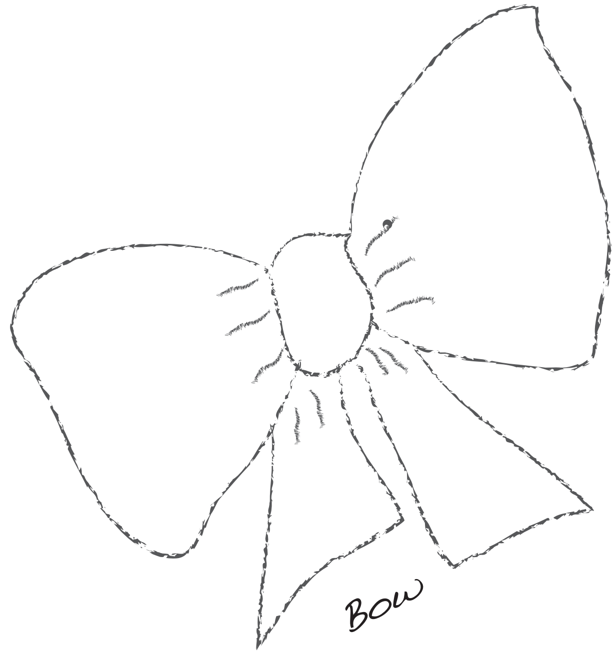
1) Cell Phone
2) Pineapple
3) Salmon
4) Cupcake
5) Tentacle
6) Bow

Part:

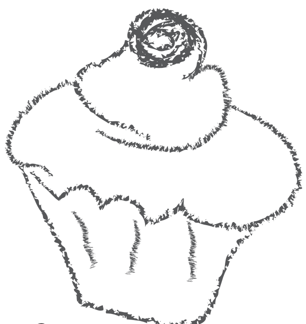
1) Head
2) Tail
3) Hand/Foot/Paw/Hoof
4) Belly
5) Nose
6) Back



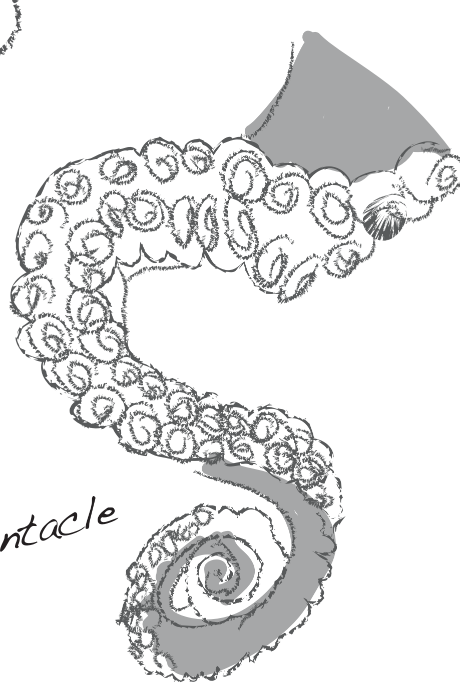
Cell Phone



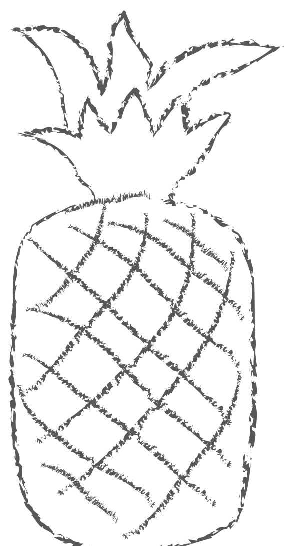
Bow



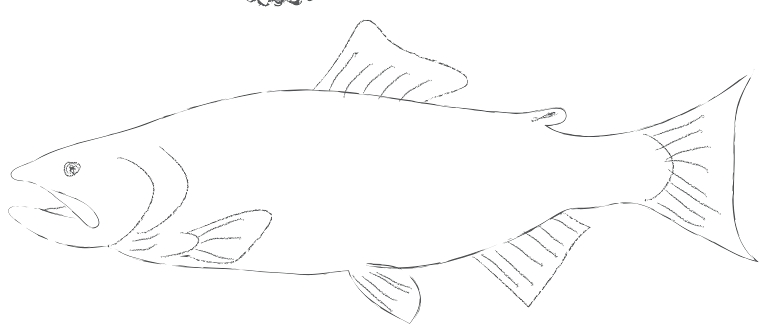
Cupcake



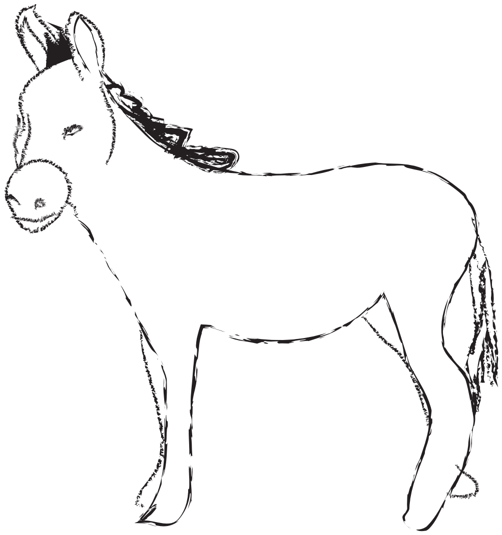
Tentacle

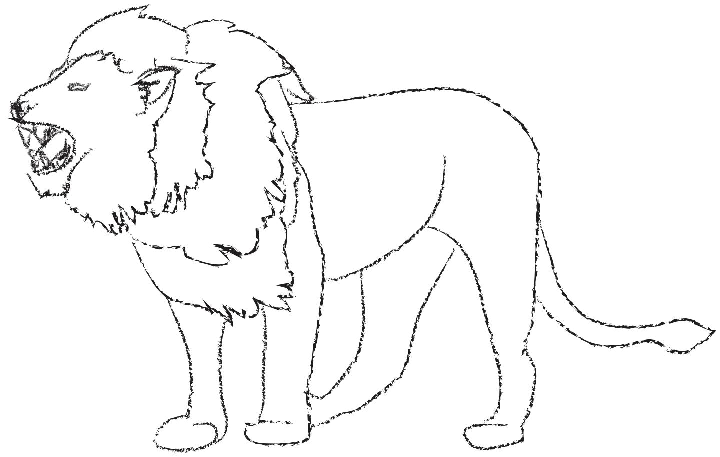


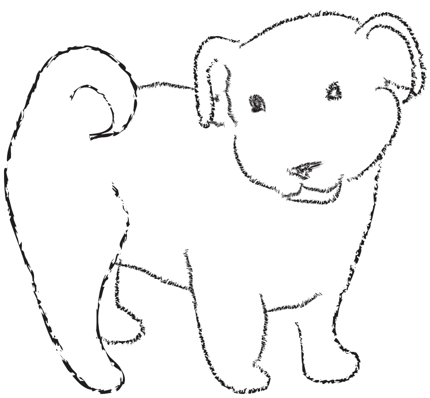
Pineapple

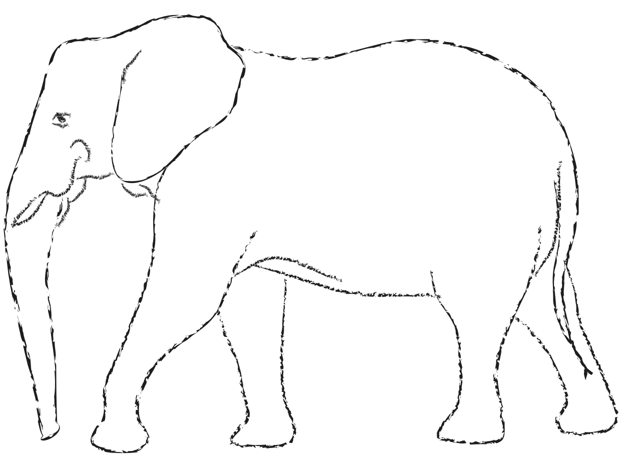


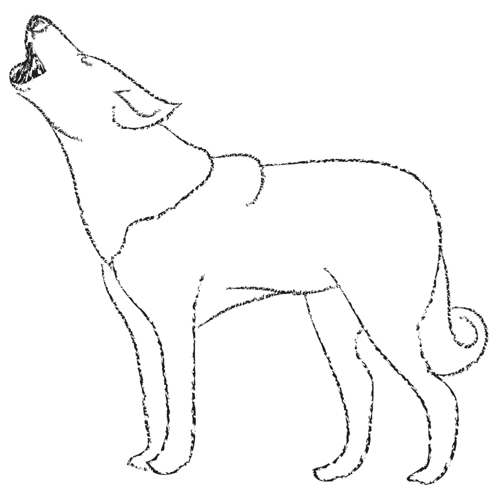
Salmon











Name: ______________________ Date: ______________

U

Unplugged

Computational Thinking

Lesson Assessment

Look at the problems below. Circle the matching sections and underline the places where there are differences. Once you've done that, write a template to create more phrases with the same pattern.

The first one has been done for you.

---

1) Triangles have three sides. Squares have four sides.

_____ have _____ sides.

2) It's fun to read books. It's fun to read magazines.

3) I love my cat's whiskers. I love my dog's tail.

I love my horse's tail. I love my cat's tail.

4) There is a cloud in the sky that looks like a dragon.

There is a leaf in the water that looks like a heart.

There was a rock in the yard that looks like a heart.

2

Course 3 | Lesson 2

Maze

Lesson time: 30 Minutes

LESSON OVERVIEW

This course is a review of maze concepts from Course 2. Students will help the zombie get to the sunflower using a combination of sequences and loops.

LESSON OBJECTIVES

Students will:

- Create a program for a given task using sequential steps
- Count the number of times an action should be repeated and represent it as a loop
- Analyze a problem and complete it as efficiently as possible
- Employ a combination of sequential and looped commands to reach the end of a maze

GETTING STARTED

Introduction

Review with students the basic maze navigation, particularly:

- Moving forward
- Turning left/right
- Looping

ACTIVITY

Maze

As your students work through the puzzles, observe how they plan the path for the zombie. Identify different strategies used and ask students to share with the whole class. This helps students to recognize that there are many ways to approach these problems. You may want to go through a few puzzles on the projector. While doing this you can ask a one student to trace the path on the screen while another writes the directions on a whiteboard.

EXTENDED LEARNING

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

Create Your Own

In small groups, let students design their own mazes and challenge each other to write programs to solve them. For added fun, make life size mazes with students as the zombie and flower.

3

Course 3 | Lesson 3

Artist

Lesson time: 30 Minutes

LESSON OVERVIEW

This course is a review of artist concepts from Course 2. Students will draw images with the artist using a combination of sequences and loops.

LESSON OBJECTIVES

Students will:

- Count the number of times an action should be repeated and represent it as a loop
- Divide the number of degrees in a circle into even segments
- Calculate the angles in equilateral and 30 60 90 triangles
- Given a perimeter and one side of a rectangle, calculate the remaining sides
- Identify symmetrical shapes
- Decompose a shape into its smallest repeatable sequence

GETTING STARTED

Introduction

Review with students the basic maze navigation, particularly:

- Moving vs Jumping
- Turning by degrees
- Looping

ACTIVITY

Artist

In the Artist levels students are not constrained to 90 degree angles. Having protractors available can be help students better visualize the angles they need.

EXTENDED LEARNING

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

The Copy Machine

- Give students two pieces of paper
- On one sheet have the students draw a simple image
- On the second sheet draw instructions for recreating that image commands to move straight and turn at various angles.
- Trade instruction sheets and attempt to recreate the image using only the provided instructions.

U

UNPLUGGED

Functional Suncatchers

Lesson time: 20 Minutes   Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

In this lesson, students will make a suncatcher out of string, beads, and a special charm. The students will follow a series of repetitive steps, then be asked to identify certain sets of “skills” that are duplicated several times. Once those skills are defined, they will be called from a main program and the whole beautiful process of creation will be recorded on a single sheet of paper. The final program will be geared toward the entire class, whatever their type of string, beads, and charms. To effectively allow for this, students will need to “abstract out” the details of their specific materials and create vague terms for an individual’s supplies. This use of generic placeholders is a wonderful introduction to variables.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) See My Suncatcher

Activity: Fun-ctional Suncatchers - 20 minutes

4) Functional Suncatcher Instructions

Wrap-up - 10 minutes

5) Flash Chat - What did we learn?
6) Vocab Shmocab

Assessment - 5 minutes

7) Functional Skills Assessment

LESSON OBJECTIVES

Students will:

- Learn to find patterns in processes
- Think about an artistic task in a different way
- Interpret symbols as they relate to physical manipulatives

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- One foot of string, thread, or fishing line per student
- 2-4 beads per student
- 2-4 other accessories (buttons, hoops, spacers) per student
- One special bead, prism, button, or student-made sun charm per student
- Pens, Pencils, & Scissors
- One Skills Sheet per group
- Fun-ctional Skills Assessment for each student

For the Teacher

- Lesson Video
- This Teacher Lesson Guide
- Print one Skills Sheet per group
- Print one Fun-ctional Skills Assessment for each student
- Gather a variety of beading supplies as described above. (These can easily be found at craft warehouses, dollar stores, or even online.)

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

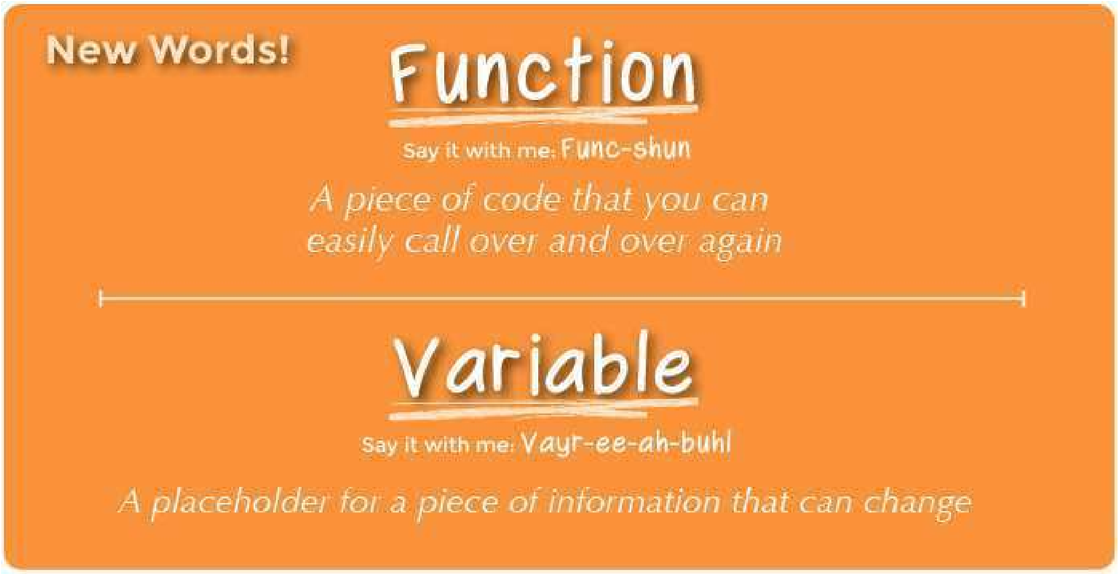
Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP *Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.*

2) Vocabulary

This lesson has two new and important words:



Function - ay it with me: unc-shun
piece of code that you can easily call over and over again

Variable - Say it with me: Vayr-ee-ah-buhl
A placeholder for a piece of information that can change

3) See My Suncatcher

Begin by holding up your example suncatcher.

- Let the class know that we will be making these today.
- Their materials may be slightly different than yours, but the steps will be pretty much the same.
    - Pointing to your bead, you can tell them that their beads may be shaped differently, or might even be a different color, but you will use the word "bead" to mean whatever it is that they ended up with.
    - Similarly, their spacers may be a different size, different texture, or maybe a different material altogether, but you will use the word "spacer" to refer to those things.
    - Finally, we will all have a "special charm." It could be a large bead, a handmade ornament, or even a random item from the room, but in all cases, we will just call it a "special charm."
    - As you share those terms, feel free to write the associated words on the board. If you want to set them equal to the items you discussed, that works well, also.

      Example:

        Bead = "Whatever style, color, or kind of bead that you have been given"

        Spacer = "A long item that is not a bead"

        Special Charm = "The crystal prism, or large glass sphere"

Now you can move in to discussing how you made your sparkling piece of art.

ACTIVITIES: (20 MIN)

4) Functional Suncatcher Instructions

- Show the students an example suncatcher
  - It's made to hang from the rearview mirror of a car, but it can also dangle from a window or a backpack zipper.
  - Let them know that they are going to make these today, and in the process, learn about programs, variables and functions.

Steps:

1) Explain to the class how the suncatcher is made.

- "First, I put a bead on the string, then I tied a knot. I put another bead on a string, and tied another knot. Then, I put a spacer on the string and tied another knot. After that, I did it all again. I put a bead on the string, then I tied a knot. I put another bead on a string, and tied another knot. Then, I put a spacer on the string and tied another knot. Finally, I put on the special charm, and tied one last knot."

2) Acknowledge that the routine takes quite a few steps, so you will provide them a sing-songy way to remember the order.

- "Bead, knot, bead, knot, spacer, knot.
  (It helps to have a bit of a rhythm with the words as you go through.)
  Bead, knot, bead, knot, spacer, knot.
  Special charm, final knot."

3) Indicate that you're going to write the steps down to this program, so that everyone has the directions in front of them.

- Use the provided "Skills Sheet" and begin to write the steps in the "Program" area as the class shouts out instructions. Write one instruction per line.
  1. Bead
  2. Knot
  3. Bead
  4. Knot

        5. Spacer
        6. Knot
- By now, the class should be noticing an issue.
    - If they don't, you can lead them to it.
        - How many instructions do we have left to go through?
        - How many lines do we have left?
        - What should we do?

4) Hand out the Skills Sheet to everyone.

SKILL 1

1) ____________________
2) ____________________
3) ____________________
4) ____________________
5) ____________________
6) ____________________

SKILL 2

1) ____________________
2) ____________________
3) ____________________
4) ____________________
5) ____________________
6) ____________________

PROGRAM

1) ______________________________
2) ______________________________
3) ______________________________
4) ______________________________
5) ______________________________
6) ______________________________

- Point out that there are two "extra" sections where they can combine steps so that you can write one name to call them all.
    - For instance, everything that you put into the top section, you can call all at once by calling "Skill 1"; this is a function.
    - Challenge the class to fill out their sheet in a way that makes sense and allows them to fit the entire sequence under the "Program" list in some way.
    - Give them about five minutes to complete their sheet (more for younger students) then listen to their suggestions.
        - There will likely be groups that need to have the process explained one-on-one.
            - Ask them what they would do if they had to repeat the sequence a thousand times or more.
    - How many different ways did the class fill out the same sheet to make the same suncatcher?

SKILL 1

1) Bead
2) Knot
3) Bead
4) Knot
5) Spacer
6) Knot

SKILL 2

1) Special Charm
2) Final Knot
3)
4)
5)
6)

PROGRAM

1) Skill 1
2) Skill 1
3) Skill 2
4)
5)
6)

5) Once you have reviewed all of the groups' solutions, hand out supplies to the students.

- Point out that different people in the class may have different supplies, and they may all be different than the ones you used.
  - Ask if that should stop them from following the program.
    - Why not?
  - We can treat the words (Bead, Spacer, Knot, etc.) as "placeholders" for whatever items we are using that fit those descriptions.
  - These are called **variables**

6) Let the students get started on their suncatchers!

WRAP-UP (5 MIN)

5) Flash Chat: What did we learn?

- What items did you use that could have been "variable" from person to person?
- How important do you think it was to have groups of skills that we could call for this lesson?
- What if we had done each set of steps 100 times instead of only twice?
- Can you think of anything else that we could group together once and call easily over and over again?
  - What about the chorus of 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.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

    "To pull on something suddenly"
    "A piece of code that you can easily call over and over again"
    "Someone who creates something that no one else has ever made"

        ...and what is the word that we learned?

ASSESSMENT (5 MIN)

7) Functional Skills 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

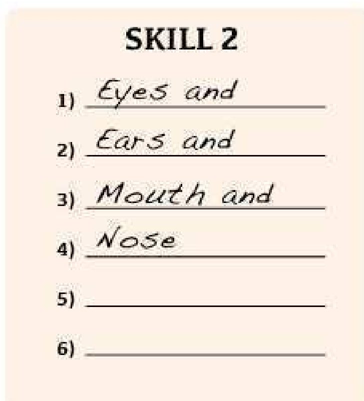
Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Functions with Song

- Did you know that the same concepts from above can be used to represent several different things?
    - Try using the Skills Sheet to represent a song, like "Head, Shoulders, Knees, and Toes."

SKILL 1

1) Head
2) Shoulders
3) Knees and
4) Toes
5) Knees and
6) Toes



SKILL 2

1) Eyes and
2) Ears and
3) Mouth and
4) Nose
5)
6)

PROGRAM

1) Skill 1
2) Skill 1
3) Skill 2
4) Skill 1
5)
6)

Name: ____________________ Date: ____________

U

Unplugged

Functional Instructions

Learning Variables and Functions through Craft

Sometimes you want to do certain instructions over and over again. That's where functions come in handy! Group all repeated instructions into one place, give them a simple name, then you can call that entire group at the same time just by using the name you gave it.

What if you want to do something over and over, but don't know what supplies you'll be working with ahead of time? This is the perfect place for variables! Variables are just placeholder words that you can put into your program so that you know where your *actual* supplies are supposed to go, once you know what they are.

Directions:

1) Take a program that contains several sets of identical instructions.

2) Move one or more of the sets of identical instructions into the "Skills" areas of the Skills Sheet.

3) Rewrite the original program, using the skill names instead of actually writing out the group of instructions that the skills describe.

New Words!

Function

Say it with me: Func-shun

A piece of code that you can easily call over and over again

Variable

Say it with me: Vayr-ee-ah-buhl

A placeholder for a piece of information that can change

Name: ______________________ Date: ______________

U

Unplugged

Functional Instructions

Skills Sheet

CODE

SKILL 1

1) ____________________
2) ____________________
3) ____________________
4) ____________________
5) ____________________
6) ____________________

SKILL 2

1) ____________________
2) ____________________
3) ____________________
4) ____________________
5) ____________________
6) ____________________

PROGRAM

1) ______________________________
2) ______________________________
3) ______________________________
4) ______________________________
5) ______________________________
6) ______________________________

Name: ______________________ Date: ______________

U

Unplugged

Fun-ctional Skills

Functions and Variables Assessment

CODE

Below, you will find three sets of skills, and a program that calls them.

Use the New Program and the skills that go with it to figure out what the steps of the Original Program were. Fill out the steps of the Original Program appropriately.

ORIGINAL PROGRAM

1) ____________
2) ____________
3) ____________
4) ____________
5) ____________
6) ____________
7) ____________
8) ____________
9) ____________
10) ____________
11) ____________
12) ____________
13) ____________
14) ____________

NEW

SKILL 1

1) banana
2) face
3) smells
4) ____________
5) ____________

SKILL 2

1) cat
2) ____________
3) ____________
4) ____________
5) ____________

SKILL 3

1) one
2) stinky
3) ____________
4) ____________
5) ____________

NEW PROGRAM

1) Skill 3
2) Skill 2
3) Skill 3
4) Skill 1
5) Skill 2

5

Course 3 | Lesson 5

Artist: Functions

Lesson time: 30 Minutes

LESSON OVERVIEW

As an introduction to functions (reuseable blocks of code), students will use functions to draw shapes on the screen and modify those functions to fit different purposes.

LESSON OBJECTIVES

Students will:

- Use a pre-determined function to draw an image with repeated features
- Modify an existing function to draw a different shape
- Distinguish between functions and loops
- Create a program that calls a function from within a loop

GETTING STARTED

Introduction

Reflect with students on the Functional Suncatchers unplugged activity

- What functions did we use to make suncatchers?
- Why did we use functions?
- Now we're going to use functions to make drawing cool images even easier!

ACTIVITY

Artist: Functions

Some students may struggle with the difference between the function definition (the actual code of the function) and the function call (used when we tell functions to run). As a visual cue we've place the function definition inside a grey box - you can equate this to the separate sheet they used to write suncatcher functions.

EXTENDED LEARNING

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

Shape Machines

Ask students to write functions on paper for simple shapes, then create a list of instructions using those functions to draw a picture.

6

Course 3 | Lesson 6

Bee: Functions

Lesson time: 30 Minutes

LESSON OVERVIEW

Using the Bee environment, students use and modify functions to help the bee collect nectar and make honey.

LESSON OBJECTIVES

Students will:

- Use a pre-determined functions to complete commonly repeated tasks
- Modify an existing function to complete a different task
- Create a function from scratch

GETTING STARTED

Introduction

ACTIVITY

Bee: Functions

EXTENDED LEARNING

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

Functionalize Your Classroom

- Brainstorm a list of daily classroom activities with students (turning in work, lining up for recess, eating lunch, etc.)
- Break students into small groups and assign each group an activitiy and a big sheet of paper
- Each group will write the daily activity as a function and create a poster with that function's "code"
- As a whole class you can then write a program for a typical day in class, calling each group's function as necessary

7

Course 3 | Lesson 7

Bee: Conditionals

Lesson time: 30 Minutes

LESSON OVERVIEW

In the Bee environment, students write programs with conditional statements. Students originally learned this concept in Course 2, but this lesson introduces more complex implementations of conditionals.

LESSON OBJECTIVES

Students will:

- Compare properties and values using >, =, < symbols
- Translate spoken language conditional statements into a program
- Execute an algorithm with a conditional statement
- Use conditional statements to make logic-based choices
- Nest conditionals to analyze multiple value conditions using if, else if, else logic
- Write functions that execute nested conditionals

GETTING STARTED

Introduction

Review conditional statements with students.

- What is a conditional statement?
- When are they useful?
- What conditional did we use in the Course 2 Bee Conditionals?

ACTIVITY

Bee: Conditionals

In the Course 2 Bee Conditionals, we only looked at simple conditionals called "if statments," such as "if there is one nectar, collect it." Bascially, we are saying if a statement is true, do something. In this stage we are going to look at what to if that statement is *not* true, we call these "if, else" statements.

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:
  - If the suit is clubs
  - If 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?

8

Course 3 | Lesson 8

Maze: Conditionals

Lesson time: 30 Minutes

LESSON OVERVIEW

Using the Maze environment, students write programs using conditionals.

LESSON OBJECTIVES

Students will:

- Compare properties and values using >, =, < symbols
- Translate spoken language conditional statements into a program
- Execute an algorithm with a conditional statement
- Use conditional statements to make logic-based choices
- Nest conditionals to analyze multiple value conditions using if, else if, else logic
- Write functions that execute nested conditionals

GETTING STARTED

Introduction

Similar to the last previous conditionals stage, only this time we're helping the zombie reach the sunflower.

ACTIVITY

Maze: Conditionals

The conditionals in this stage pertain to whether or not there is a path available to the right, left, or ahead. If there is chomper in the way, the conditional will report that there is no path available in that direction.

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:
    - If the suit is clubs
    - If 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?

U

UNPLUGGED

Songwriting

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON 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.

TEACHING SUMMARY

Getting Started - 20 minutes

1) Review
2) Vocabulary
3) Sing a Song

Activity: Songwriting - 20 minutes

4) Songwriting

Wrap-up - 5 minutes

5) Flash Chat - What did we learn?
6) Vocab Shmocab

Assessment - 5 minutes

7) Songwriting Assessment

LESSON OBJECTIVES

Students will:

- Locate repeating phrases inside song lyrics
- Identify sections of a song to pull into a function (chorus)
- Describe how functions can make programs easier to write

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Songwriting Worksheet
- Pens & Pencils

For the Teacher

- Lesson Video
- This Teacher Lesson Guide
- Print several Songwriting Worksheets for each group
- Print one Songwriting Assessment for each student.
- Access to the internet, or pre-downloaded songs and lyrics for activity

GETTING STARTED (20 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. You can do this as one large group or have students discuss with an elbow partner.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) 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

3) 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"

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 more chance.
Then I'll turn you into a goon!
The next day...

Chorus

"I gave you 3 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 the some of the same reasons that it is helpful to songwriters. - 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!

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.

ACTIVITIES: (20 MIN)

4) Songwriting

- 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?
- Sing your choruses together to find out!

Play this game over and over until the class has little trouble identifying the choruses.

- 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.

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

WRAP-UP (5 MIN)

5) 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.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

  "A piece of code that you can call over and over again"
  "A baby horse"
  "Getting help from a large group of people to finish something faster"

  ...and what is the word that we learned?

ASSESSMENT (5 MIN)

7) 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

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

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

CODE

Song 1 Name:

Chorus:

Song 2 Name:

Chorus:

U

UNPLUGGED

Dice Race

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

In this lesson, students will relate the concept of algorithms back to everyday 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.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) What We Do Daily

Activity: Real-Life Algorithms - 20 minutes

4) Real-Life Algorithms: Dice Race

Wrap-up - 5 minutes

5) Flash Chat - What did we learn?

Assessment - 10 minutes

6) Daily Algorithms

LESSON OBJECTIVES

Students will:

- Name various activities that make up their day
- Decompose large activities into a series of smaller events
- Arrange sequential events into their logical order

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Dice (1 per pair)
- Pens/Pencils/Markers
- Real-Life Algorithms Worksheet: Dice Race
- Assessment Worksheet: Daily Algorithms

For the Teacher

- Lesson Video
- Teacher Lesson Guide
- Print one Real-Life Algorithms Worksheet per group
- Print one Assessment Worksheet: Daily Algorithms per student

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP *Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.*

2) 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

3) 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, 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!

ACTIVITY: (20 MIN)

4) Real-Life Algorithm Worksheet: Dice Race

LESSON TIP *You know your classroom best. As the teacher, decide if students should do this in pairs or small groups.*

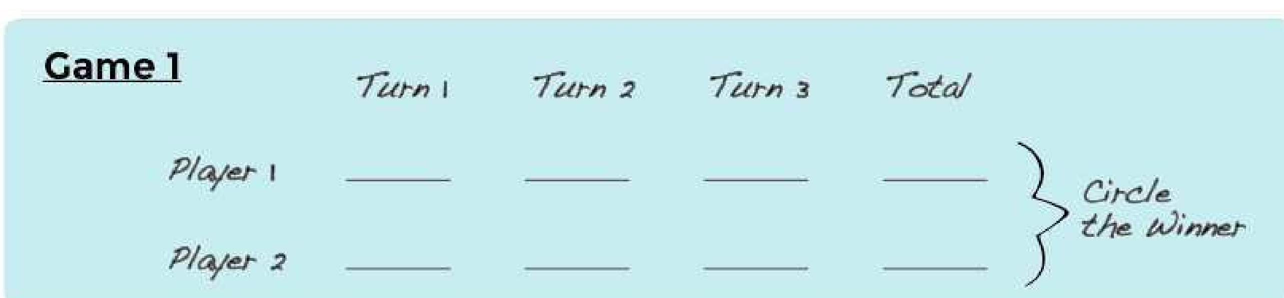
- You can use algorithms to help describe things that people do every day. In this activity, we will create an algorithm to describe how we play the Dice Race Game.
- The hardest part about getting a problem ready for a computer can be figuring out how to describe real-life activities. We're going to get some practice by playing and describing the Dice Race game.

Directions:

1. Read the rules below.
2. 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:

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



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.*

WRAP-UP (5 MIN)

5) 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?

ASSESSMENT (15 MIN)

6) Assessment Worksheet: Daily Algorithms

- 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.

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.

Created in partnership with THINKERSMITH®

Name: ______________________ Date: ______________

U

Unplugged

Real-Life Algorithms

Dice Race Activity

CODE

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	______	______	______	______	

U

Unplugged

Dice Race

Assessment Worksheet

C O D E

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 : ______________________________

11

Course 3 | Lesson 11

Artist: Nested Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Students use the Artist environment to write programs that have looped statements inside another loop, which is called a nested loop.

LESSON OBJECTIVES

Students will:

- Count the number of times an action should be repeated and represent it as a loop
- Divide the number of degrees in a circle into even segments
- Given a number of segments, calculate the degrees need to complete a circle
- Break complex tasks into smaller repeatable sections
- Combine simple shapes into complex designs with nested loops

GETTING STARTED

Introduction

ACTIVITY

Artist: Nested Loops

EXTENDED LEARNING

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

12

Course 3 | Lesson 12

Farmer: While Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Using while loops, students control a farmer shovel dirt into holes until they're full and remove dirt from piles until it's all gone.

LESSON OBJECTIVES

Students will:

- Distinguish between loops that repeat a fixed number of times and loops that repeat until a condition is met
- Use a while loop to create programs that can solve problems with unknown values

GETTING STARTED

Introduction

ACTIVITY

Farmer: While Loops

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!

13

Course 3 | Lesson 13

Bee: Nested Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Students use the Bee environment to write programs using nested loops.

LESSON OBJECTIVES

Students will:

- Break a sequence of steps into a hierarchy or looped sequences
- Nest loops and conditionals to analyze multiple value conditions using if, else if, else logic

GETTING STARTED

Introduction

ACTIVITY

Bee: Nested Loops

EXTENDED LEARNING

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

14

Course 3 | Lesson 14

Bee: Debugging

Lesson time: 30 Minutes

LESSON OVERVIEW

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 missing blocks, extra blocks, and misordered blocks.

LESSON OBJECTIVES

Students will:

- Predict where a program will fail
- Modify an existing program to solve errors
- Identify an algorithm that is unsuccessful when the steps are out of order
- Reflect on the debugging process in an age-appropriate way

GETTING STARTED

Introduction

Ask students about problems they solve in everyday life.

- How do you fix something that isn't working?
- Do you follow a specific series of steps?
- The puzzles in this unit have already been solved for you (yay!), but they don't seem to be working (boo!)
- We call the problems in these programs "bugs," and it will be your job to "debug" them.
- Brainstorm different ways to find and fix bugs.

ACTIVITY

Bee: Debugging

As your students work through the puzzles, observe how they search for bugs. Identify different strategies used and ask students to share with the whole class. This helps students to recognize that there are many ways to approach these problems. Have students follow the path described by the code with their fingers to find potential bugs.

EXTENDED LEARNING

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

Planting bugs

Have students go back through previous levels, purposfully adding bugs to their solutions. They can then ask other students to debug their work. This can also be done with paper puzzles.

15

Course 3 | Lesson 15

Bounce

Lesson time: 30 Minutes

LESSON OVERVIEW

In this special level students get to build their own Flappy Bird clone. By using event handlers to detect mouse clicks and object collisions.

LESSON OBJECTIVES

Students will:

- Match blocks with the appropriate event handler
- Create a game using event handlers
- Share a creative artifact with other students

GETTING STARTED

Introduction

LESSON TIP

Students will have the opportunity to share their final product with a link. This is a great opportunity to show your school community the great things your students are doing. Collect all of the links and keep them on your class website for all to see!

ACTIVITY

Bounce

In the final stage of this lesson students are able to tweak their game to make it unique - encourage them to see how different they can make each game within the constraints provided.

EXTENDED LEARNING

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

Look Under the Hood

When you share a link to your game, you also share all of the code that goes behind it. This is a great way for students to learn from each other.

- Post links to completed games online or on the board.
  - Make a game of your own to share as well!
- When students load up a link, have them click the "How it Works" button to see the code behind the game.
- Discuss as a group the different ways your classmates coded their games.
  - What suprised you?
  - What would you like to try?
- Choose someone else's game and build on it. (Don't worry; the original game will be safe.)

16

Course 3 | Lesson 16

Play Lab: Create a Story

Lesson time: 30 Minutes

LESSON OVERVIEW

In this activity, students will have the opportunity to apply all of the coding skills they've learned to create an animated story. It's time to get creative and create a story in the Play Lab!

LESSON OBJECTIVES

Students will:

- Identify actions that correlate to input events
- Create an animated, interactive story using sequence, loops, and event-handlers
- Share a creative artifact with other students

GETTING STARTED

Introduction

Review the event handling students did in Bounce:

- What did events did you use in coding Flappy?
- Now you're going to animate multiple characters using events triggered by the arrow keys to tell a story.
- This is your chance to get really creative!

LESSON TIP

Students will have the opportunity to share their final product with a link. This is a great opportunity to show your school community the great things your students are doing. Collect all of the links and keep them on your class website for all to see!

ACTIVITY

Play Lab: Create a Story

his is one of the most free-form plugged activities of the course. At the final stage students have the freedom to create a story of their own. You may want to provide structured guidelines around what kind of story to write, partiularly for students who are overwhelmed by too many options.

EXTENDED LEARNING

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

Look Under the Hood

When you share a link to your story, you also share all of the code that goes behind it. This is a great way for students to learn from each other.

- Post links to completed stories online or on the board.
  - Make a story of your own to share as well!
- When students load up a link, have them click the "How it Works" button to see the code behind the story.
- Discuss as a group the different ways your classmates coded their stories.
  - What suprised you?
  - What would you like to try?
- Choose someone else's story and build on it. (Don't worry; the original story will be safe.)

17

Course 3 | Lesson 17

Play Lab: Create a Story

Lesson time: 30 Minutes

LESSON OVERVIEW

Building on the previous Play Lab activity, students will add deeper interactivity as they build their own video games.

LESSON OBJECTIVES

Students will:

- Identify actions that correlate to input events
- Create an animated, interactive story using sequence, loops, and event-handlers
- Share a creative artifact with other students

GETTING STARTED

Introduction

LESSON TIP

Students will have the opportunity to share their final product with a link. This is a great opportunity to show your school community the great things your students are doing. Collect all of the links and keep them on your class website for all to see!

ACTIVITY

Play Lab: Create a Game

Designing a game that is fun to play yet challenging enough to avoid boredom is a tough task. Encourage students to consider how their programming choices make different aspects of their game harder, easier, frustrating, and fun.

EXTENDED LEARNING

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

Look Under the Hood

When you share a link to your game, you also share all of the code that goes behind it. This is a great way for students to learn from each other.

- Post links to completed games online or on the board.
  - Make a game of your own to share as well!
- When students load up a link, have them click the "How it Works" button to see the code behind the game.
- Discuss as a group the different ways your classmates coded their stories.
  - What suprised you?
  - What would you like to try?
- Choose someone else's game and build on it. (Don't worry; the original game will be safe.)

Created in partnership with THINKERSMITH®

U

UNPLUGGED

The Internet

Lesson time: 15 Minutes   Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

In this lesson, students will pretend to flow through the Internet, all the while learning about Internet connections, URLs, IP Addresses, and the DNS.

TEACHING SUMMARY

Getting Started - 25 minutes

1) Review
2) Vocabulary
3) Getting the Message

Activity: The Internet - 15 minutes

4) The Internet

Wrap-up - 10 minutes

5) Flash Chat - What did we learn?
6) Vocab Shmocab

Assessment - 5 minutes

7) The Internet Assessment

LESSON OBJECTIVES

Students will:

- Learn about the complexity of sending messages over the Internet
- Translate URLs into IP Addresses
- Practice creative problem solving

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Paper for writing messages to send
- Paper for creating a sign to "label" each "server"
- Pens & Pencils
- IP Address and Delivery Type Cards Found Here

For the Teacher

- Lesson Video
- This Teacher Lesson Guide
- Print enough IP Address Cards and Delivery Type Cards for each group to draw
- Print one Internet Assessment for each student
- Access to the Internet (such as getip.com) or 6+ pre-determined URL/IP address combinations

GETTING STARTED (20 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) Vocabulary

This lesson has several new and important words:

New 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 conneced to each other

Servers
Say it with me: Serv-ers

Computers that exist only to provide things for 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

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

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

LESSON TIP

confusing to do a brief intro to the words as a "see if you can spot these during the day" type of heads-up.

3) 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 is.
  - 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 is like for me to send the 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!

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.

With that said, let's try to understand what the DNS does by making a little DNS table ourselves.

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	www.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 getip.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!

ACTIVITIES: (20 MIN)

4) The Internet

Directions:

1) Create your own DNS table, similar to what is shown above.

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 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.

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.

Info:

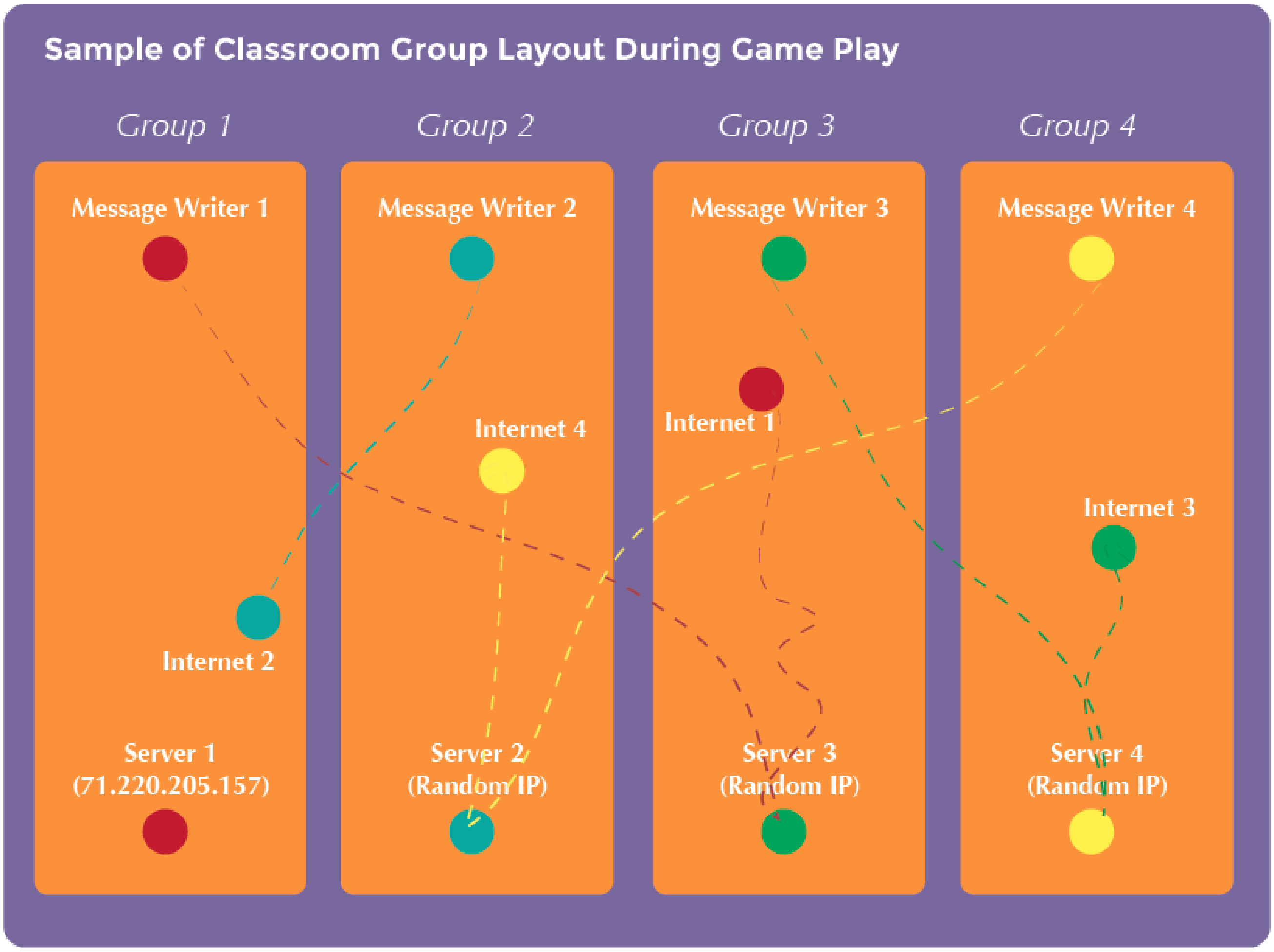
1) **Wi-Fi:** Convenient, but spotty. Wi-Fi doesn't require cables, but since the signal bounces all over the place, packets can get lost pretty easily.
Simulation: Internet must carry each packet on their shoulder (no hands).

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.

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.

- 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:



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.

WRAP-UP (5 MIN)

5) 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.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

  "A piece of code that you can call over and over again"
  "Computers that exist only to provide information to others"
  "The bottom leg of a triangle"

...and what is the word that we learned?

- What was your favorite new word?
- Is there a word that we learned today whose meaning you remember, but don't remember the word?
- Is there a word that we learned where you remember the word, but not what it means?

ASSESSMENT (5 MIN)

7) The 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.

Created in partnership with THINKERSMITH®

Internet

How the Internet Does What it Does

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.

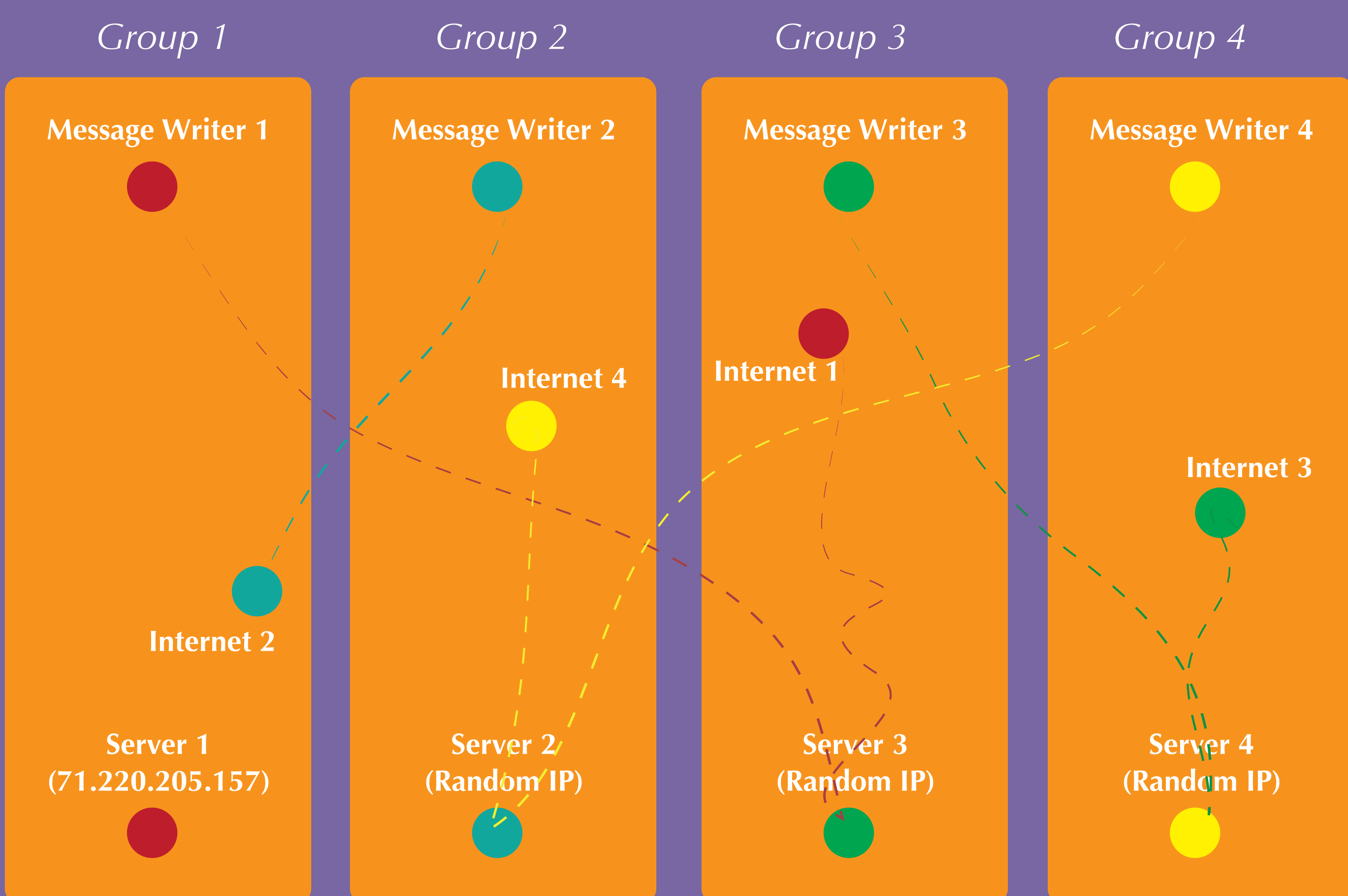
Internet

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

Internet

Delivery Type Cards

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

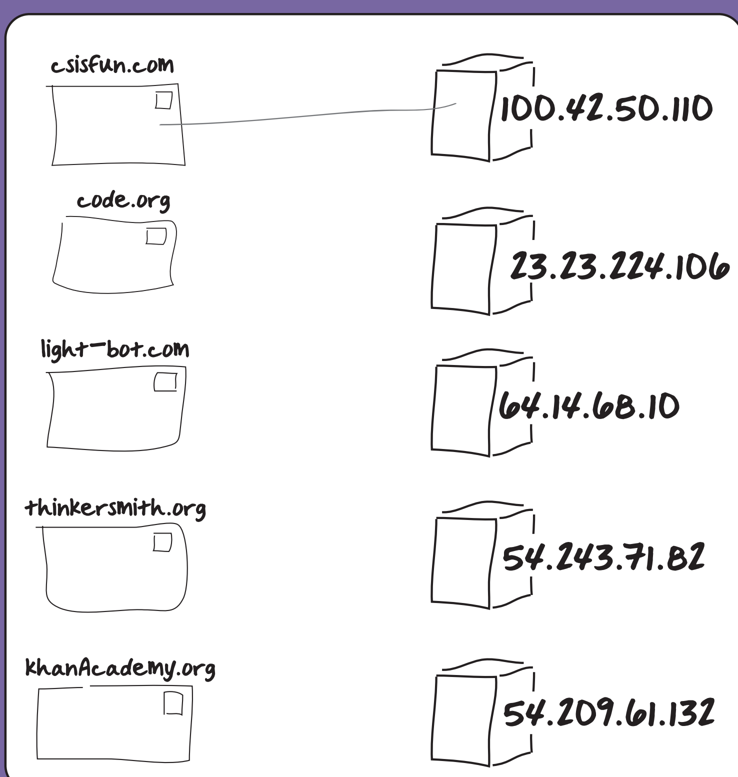
How the Internet Does What it Does

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.

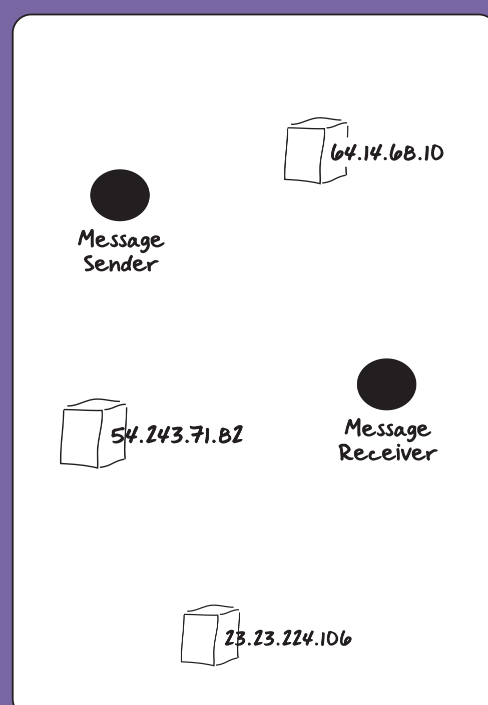
DNS Look-Up Table

#	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.



U

UNPLUGGED

Crowdsourcing

Lesson time: 20 Minutes   Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

In computer science, we face some big, daunting problems. Challenges like finding large prime numbers or sequencing DNA are almost impossible to do alone. Adding the power of others makes these tasks managable. This lesson will show your students how helpful teamwork can really be.

TEACHING SUMMARY

Getting Started - 20 minutes

1) Review
2) Vocabulary
3) Big Problems

Activity: Crowdsourcing - 20 minutes

4) Working Together

Wrap-up - 5 minutes

5) Flash Chat - What did we learn?
6) Vocab Shmocab

LESSON OBJECTIVES

Students will:

- 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

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Playing Cards (1 Deck per group)

For the Teacher

- Lesson Video

Playing cards usually come in multi-packs at the dollar store, but if you're rushed you can print some from online onto

MATERIALS TIP

cardstock and cut them out with a slicer.

For the Teacher

- This Teacher Lesson Guide
- Crowdsourcing Activity:Working Together
- Jar of lots of something (Pennies, Buttons, Slips of Paper)

GETTING STARTED (20 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) 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

3) Big Problems

- 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.*

ACTIVITIES: (20 MIN)

4) Crowdsourcing Activity:Working Together

- 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 (5 MIN)

5) Flash Chat: What did we learn?

- Have you ever tried to sort a pile of cards by yourself before?
- Do you think it was easier or harder to have help?
- What other things do you have to do sometimes that would 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.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

  "Pulling something out"
  "Getting help from a large group of people to finish something faster"
  "Something that is not cooked"

  ...and what is the word that we learned?

ASSESSMENT (0 MIN)

7) No Indivdual Assessment

- Since crowdsourcing is not meant to be done alone, there is no individual assessment.
- The final assessment of this lesson is the result of the previous activity.

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.

U

UNPLUGGED

Digital Citizenship

Lesson time: 30 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

In collaboration with Common Sense Media, 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.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) Personal vs. Private Online

Activity: Digital Citizens - 30 minutes

4) Digital Citizens

Wrap-up - 10 minutes

5) Flash Chat - What did we learn?
6) Vocab Shmocab

Assessment - 5 minutes

7) Digital Citizens Assessment

LESSON OBJECTIVES

Students will:

- 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

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Papercraft Super Heroes

- Pens, Pencils, & Scissors
- Super Digital Citizen Assessment for each student

For the Teacher

- Lesson Video
- This Teacher Lesson Guide
- A good selection of male and female Papercraft Super Heroes sheets for the whole class
- Print one Super Digital Citizen Assessment for each student

GETTING STARTED (15 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we had had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP

Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) Vocabulary

This lesson has one new and important word:

Digital Citizen - Say it with me: Dih-jih-tal Sit-i-zen
Someone who acts safely, responsibly, and respectfully online

3) 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:

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

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.

- 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.

ACTIVITIES: (20 MIN)

4) Papercraft Super Heroes

- 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:

1) Have each student grab a small selection of papercraft sheets and encourage them to blend the pieces to make their very own super hero.

2) Allow plenty of time for students to cut, glue, and color.

3) Give students a 5 minute warning to wrap up.

4) 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!

5) 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 Media](#) page on Scope and Sequence.

WRAP-UP (5 MIN)

5) 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.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

"One who is afraid of spiders"
"A baby horse"
"Someone who acts safely, responsibly, and respectfully online"

...and what is the word that we learned?

ASSESSMENT (5 MIN)

7) 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 Media

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

Created in partnership with THINKERSMITH®

Using Materials From common sense

Name: ______________________ Date: ______________

U

Unplugged

Digital Citizenship

Assessment Worksheet

CODE

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.

21

Course 3 | Lesson 21

Artist: Patterns

Lesson time: 30 Minutes

LESSON OVERVIEW

Students use the Artist environment to explore nested loops that draw complex patterns.

LESSON OBJECTIVES

Students will:

- Count the number of times an action should be repeated and represent it as a loop
- Divide the number of degrees in a circle into even segments
- Given a number of segments, calculate the degrees need to complete a circle
- Break complex tasks into smaller repeatable sections
- Combine simple shapes into complex designs with nested loops
- Compose complex patterns from multiple nested loops

GETTING STARTED

Introduction

In this stage, students will explore how nesting loops several levels deep can allow them to create complex and beautiful repeating patterns. There are no assessments at the end of this stage, it is a final challenge to encourage students to push themselves.

ACTIVITY

Artist: Patterns

EXTENDED LEARNING

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

Paper Pattern Building

Pair students, and assign each pair a basic geometric shape to examine. For that shape, they will do the following:

- Write instructions to draw that shap using a loop
- Write instructions to draw a row of that shape by repeating the previous loop
- Write instructions to draw a complex pattern by repeating the previous loop
- Use those instructions to walk another student through drawing that same pattern

Course 4

OVERVIEW

Course 4 is designed for students who have taken Courses 2 and 3. Students will learn how to tackle puzzles with increased complexity as they learn how to combine several concepts when solving each challenge. By the time this Course is completed, students will be creating programs that let them showcase multiple skills, including for loops and functions with parameters. Recommended for grades 4-8.

Lesson Sequence of Course 4

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

#	Lesson Name	Description
1	**Algorithms: Tangrams**	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 more detail.
2	Maze and Bee	First help the zombie get to the sunflower using a combination of sequences and loops, then review conditionals with the flower-hunting bee.
3	Artist: Loops Review	Use the traditional artist character, as well as a very artistic zombie, to complete and recreate repetitive drawings.
4	**Variables in Envelopes**	Explains what variables are and how we can use them in many different ways.
5	**Abstraction with Mad Glibs**	Analyze stories for differences so that they can abstract them away. Those abstracted stories become templates for fun and crazy new ones.
6	Artist: Variables	Explore the creation of repetitive designs using variables.
7	Play Lab: Variables	Play with variables in a situation that illustrates just how useful they can be.
8	**For Loop Fun**	Learn how to use loops with extra structure built right in.

#	Lesson Name	Description
9	Bee: For Loops	Use the Bee environment to write programs that use loops with embedded counters/index variables.
10	Artist: For Loops	Use the Artist environment to write programs with for loops, similar to what they did in the previous Bee level.
11	Play Lab: For Loops	Building on the previous Play Lab activity, students will add deeper interactivity as they build their own video games.
12	Artist: Functions	Use the Artist environment to draw complicated images using functions for repeated tasks.
13	**Songwriting with Parameters**	help students intuitively understand why combining chunks of code into functions is such a helpful practice.
14	Artist: Functions with Parameters	Use the Artist environment to draw complicated images using functions with parameters to create similar shapes with small differences.
15	Play Lab: Functions with Parameters	Having experienced the creation and use of functions and parameters, students will get the opportunity to use the skill in the creation of Play Lab games.
16	Bee: Functions with Parameters	This short stage illustrates how students can use their new skills with functions and parameters to change direction using binary logic.
17	**Binary Images**	Illustrates how a computer can store even more complex information (such as images and colors) in binary, as well.
18	Artist: Binary	Build binary images, translating 0s and 1s to offs and ons (or blacks and whites).
19	Super Challenge - Variables	Dig deep into what students have learned throughout their journey to solve a handful of complex puzzles.
20	Super Challenge - For Loops	Use a mix of different environments to test their knowledge of for loops.

#	Lesson Name	Description
21	Super Challenge - Functions with Parameters	This challenging stage allows students to hone their skills with functions and parameters to solve complex puzzles with grace and efficiency.
22	Extreme Challenge - Comprehensive	The final stage in this course is intended to test comprehension and transfer of all concepts to blended puzzles.

U

UNPLUGGED

Algorithms: Tangrams

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

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 more detail.

This activity will show both options.

TEACHING SUMMARY

Getting Started - 10 minutes

1) Review
2) Vocabulary
3) Intro to Tangrams

Activity: Algorithms with Tangrams - 20 minutes

4) Algorithms

Wrap-up - 10 minutes

5) Flash Chat: What did we learn?
6) Vocab-Shmocab

Assessment - 10 minutes

7) Algorithm Assessment

LESSON OBJECTIVES

Students will:

- 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

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Tangram Set & Algorithm Card Images Pack
- Scratch paper for writing algorithms or building images

- Tangram Assessment Worksheet
- Pens/Pencils
- Scissors

For the Teacher

- Print one Algorithm Card Images Pack per group
- Print one Tangram Set per student
- Print one Tangram Assessment Worksheet per student
- Provide student with scissors, paper, pens & pencils

GETTING STARTED (10 MIN)

1) Review

This is a great time to review computer science experiences from the past. Where did everyone leave off?

Here are some questions that you can ask in review:

- Do you remember what an algorithm is?
- What does an algorithm have to do with sequence?

LESSON TIP *Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.*

2) Vocabulary

This lesson has one important word to review:

Algorithm - Say it with me: Al-go-ri-thm
A list of steps that you can follow to finish a task

3) Intro to Tangrams

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 Algorithms activity.

Usually, Tangrams are 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.

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.

ACTIVITY: ALGORITHMS (20 MIN)

4) 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.

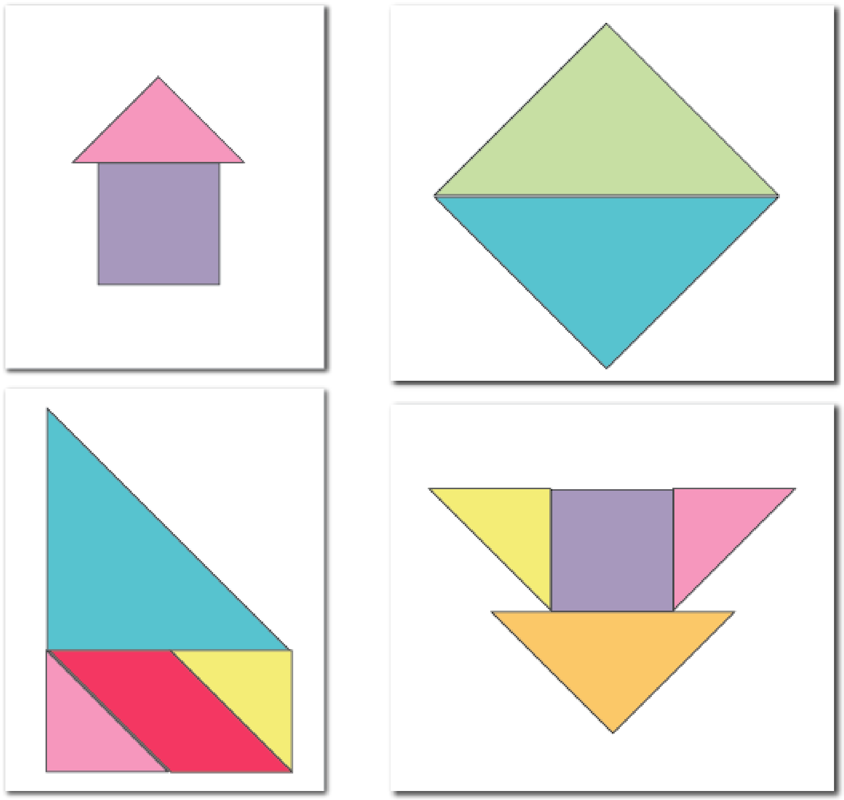
You might think that this will make it easier, but it won't, because students will also not get to actually *look* at the image that we are trying to recreate! Instead, a teammate will be *describing* the image to us.

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 (10 MIN)

5) 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?

6) Vocab Shmocab

You can choose to do these as a class, or have the students discuss with an elbow partner and share.

- Do you remember the definition of the word "algorithm"?

  "A list of steps that you can follow to finish a task"
  "An algorithm that has been coded into something that can be run by a machine"
  "Finding and fixing problems in your algorithm or program"

ASSESSMENT (10 MIN)

7) Algorithm Assessment

EXTENDED LEARNING

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

At Your Word

- Make up simple Tangram Algorithms for your class, and have them figure out how many different images they can create that follow that algorithm to the letter.
  - Choose a couple of drawings to analyze against the algorithm.

- Move toward more specific algorithms that leave little room for variation.

Shapely Debugging

- Create an algorithm for an image, and provide the class with a Tangram Arrangement that doesn't *quite* match.
  - Ask the class if the image matches the algorithm.
  - Can they figure out where it went wrong?
  - Do you need to throw the whole arrangement out and start over or can you just start from where the algorithm went wrong?

LESSON TIP

Try to focus on misplacements that allow the class to back up only a few steps to fix the algorithm. We want to get it in the students' heads that they don't have to delete entire programs if something doesn't work, they just need to find the error and fix that bit (and any bit that was relying on that instruction).

CONNECTIONS AND BACKGROUND INFORMATION

ISTE Standards (formerly NETS) Satisfied by this Lesson Include:

- 1c. Use models and simulations to explore complex systems and issues
- 2d. Contribute to project teams to produce original works or solve problems
- 4b. Plan and manage activities to develop a solution or complete a project
- 6c. Use multiple processes and diverse perspectives to explore alternative solutions

The activities in this lesson support CSTA K-12 Computer Science Standards:

- CT.L1:6-01 Understand and use the basic steps in algorithmic problem-solving
- CT.L1:6-02 Develop a simple understanding of an algorithm using computer-free exercises
- CPP.L1:6-05 Construct a program as a set of step-by-step instructions to be acted out

Next-Gen Science Standards

- 3-5-ETS1-2 Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem

Common Core Mathematical Practices

- 1. Make sense of problems and persevere in solving them
- 6. Attend to precision

Common Core Math Standards

- 3.G.A.1 Understand that shapes in different categories may share attributes and that the shared attributes can define a larger category
- 5.G.B.3 Understand that attributes belonging to a category of two-dimensional figures also belong to all subcategories of that category

Common Core Language Arts Standards

- L.3.6 Acquire and use accurately grade-appropriate conversational, general academic, and domain-specific words and phrases, including those that signal spatial and temporal relationships.
- L.4.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal precise actions, emotions, or states of being and that are basic to a particular topic.
- L.5.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal contrast, addition, and other logical relationships.

Name: ______________________ Date: ______________

U

Unplugged

Algorithms

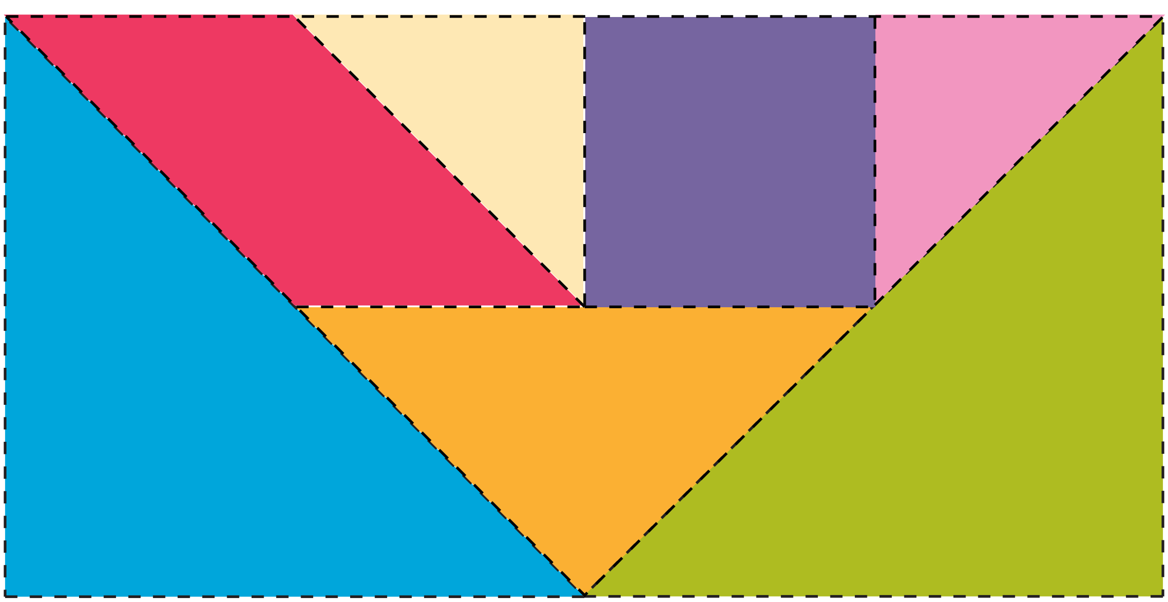
Tangrams Algorithm Activity

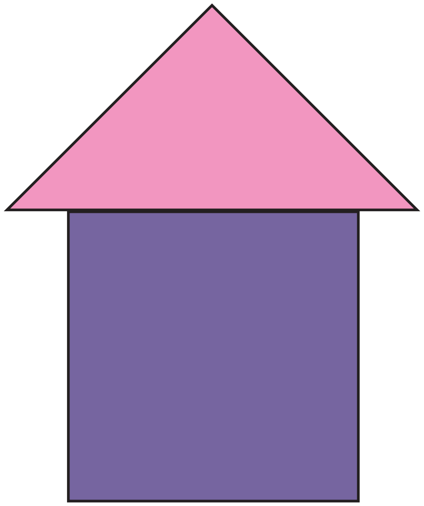
This lesson shows us something important about algorithms. If you keep an algorithm simple there are lots of ways to use it. If you want to make sure everyone ends up with the same thing, then your algorithm needs to have a lot of detail.

This activity will show both options.

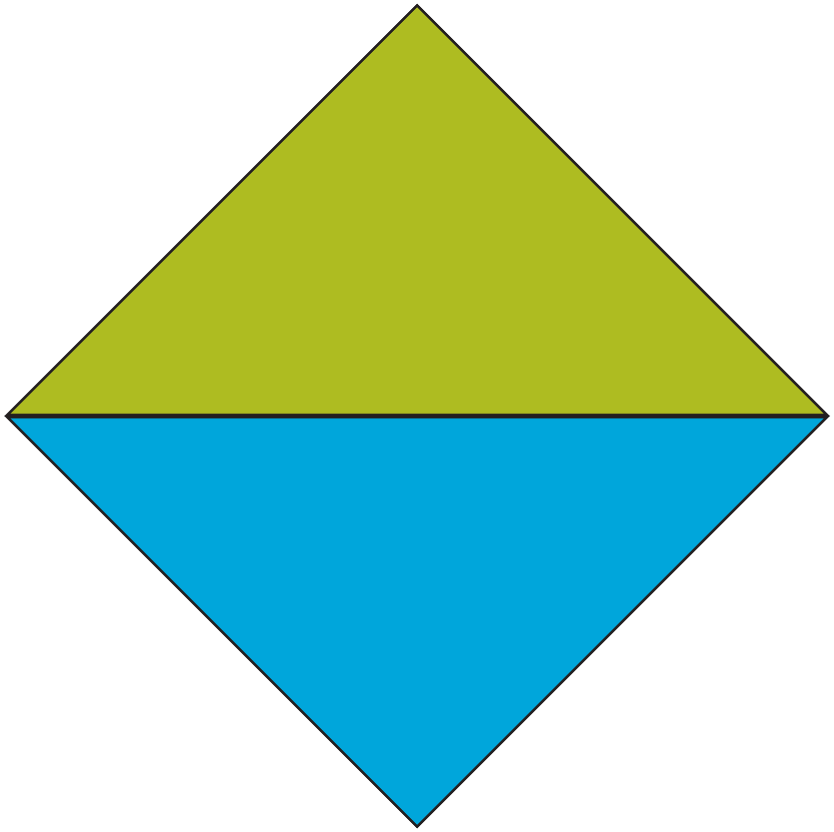
Directions:

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.





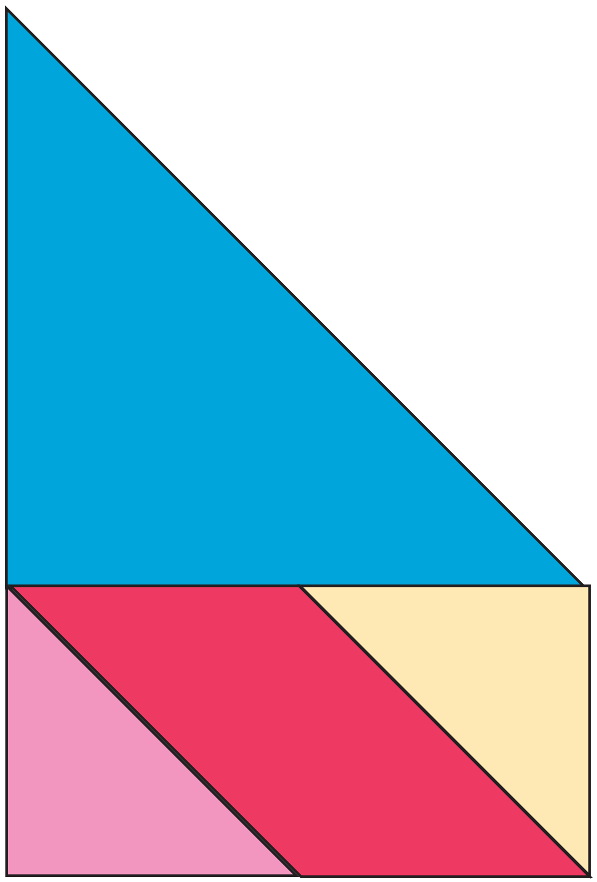
U
Algorithms
Algorithms Card 2
CODE



Revision 140929.1a

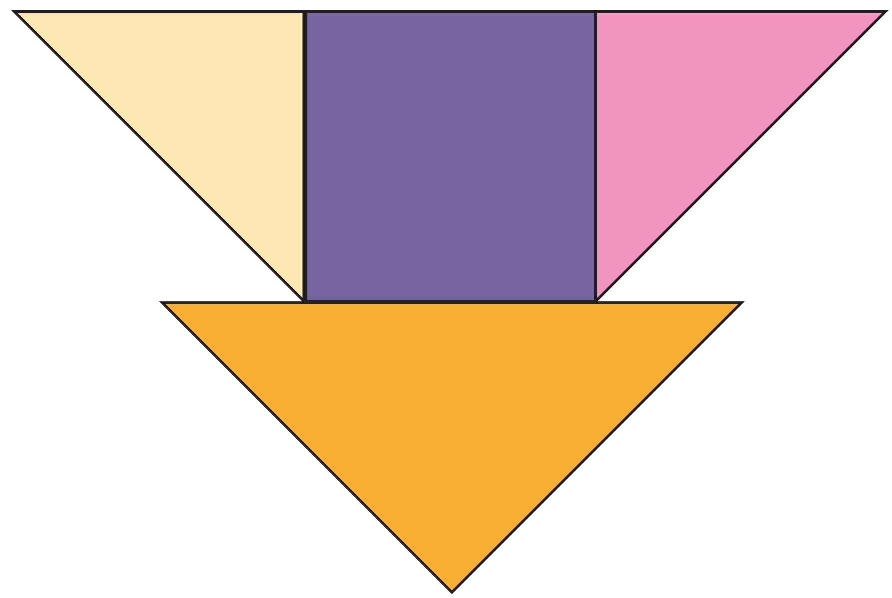
Algorithms

Algorithms Card 3



Algorithms

Algorithms Card 4



Name: ______________________ Date: ______________

U

Unplugged

Algorithms

Tangrams Assessment Worksheet

CODE

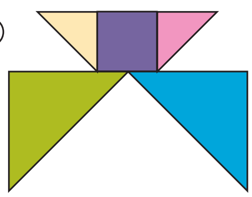
Very specific algorithms help multiple people create identical products.

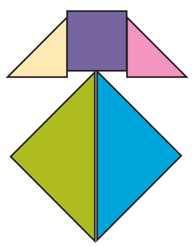
Less specific algorithms allow a great deal of flexibility for every person to have something different.

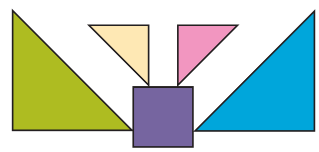
Circle the drawing that does not follow the algorithm provided.

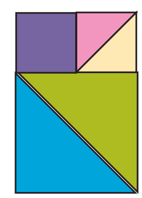
Algorithm #1

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

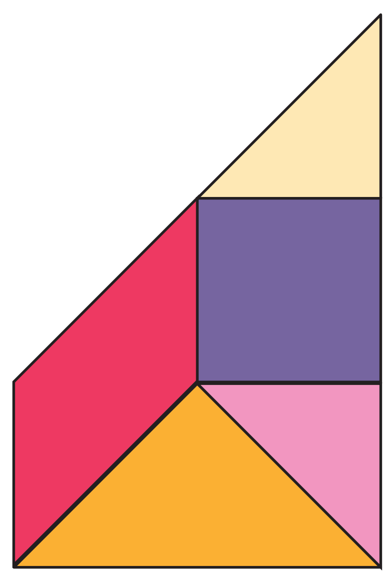
1) Use two triangles, a square, and another piece
2) Line two triangles up with the square
3) Put the last piece on top of the square

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

2

Course 4 | Lesson 2

Maze and Bee

Lesson time: 30 Minutes

LESSON OVERVIEW

This course is a review of maze concepts from Courses 2 and 3. Students will first help the zombie get to the sunflower using a combination of sequences and loops, then review conditionals with the flower-hunting bee.

LESSON OBJECTIVES

Students will:

- Create a program for a given task using sequential steps
- Count the number of times an action should be repeated and represent it as a loop
- Analyze a problem and complete it as efficiently as possible
- Employ conditional statements to assess which actions are correct for a given step

GETTING STARTED

Introduction

Review with students the basic maze navigation, particularly:

- Moving forward
- Turning left/right
- Looping
- Conditionals

ACTIVITY

Maze and Bee

As your students work through the puzzles, observe how they plan the path for the zombie or bee. Identify different strategies used and ask students to share with the whole class. This helps students to recognize that there are many ways to approach these problems. You may want to go through a few puzzles on the projector. While doing this you can ask a one student to trace the path on the screen while another writes the directions on a whiteboard.

EXTENDED LEARNING

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

Create Your Own

In small groups, let students design their own mazes and challenge using checkerboards and strips of paper. Can they recreate a bee conditionals puzzle using red and black checkers?

3

Course 4 | Lesson 3

Artist: Loops Review

Lesson time: 30 Minutes

LESSON OVERVIEW

This is a review of loops and nested loops. Students will use the traditional artist character, as well as a very artistic zombie, to complete and recreate repetitive drawings.

LESSON OBJECTIVES

Students will:

- Create programs that utilize repetition to create gorgeous designs
- Use trial and error to recreate detailed designs in proper scale
- Divide the number of degrees in a circle into even segments
- Calculate the angles in equilateral and 30 60 90 triangles
- Decompose a shape into its smallest repeatable sequence

GETTING STARTED

Introduction

Review with students the basic artist navigation, particularly:

- Moving forward
- Turning left/right
- Looping
- Angles

ACTIVITY

Artist: Loops Review

Students will see a variety of shapes in this lesson. Some of the designs get pretty intricate. Your class may find it helpful to have a protractor and paper on hand. Feel free to promote trial and error as well.

EXTENDED LEARNING

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

Paper Snowflakes

Hand your class small square sheets of paper and have them fold the papers in half then cut simple designs. How many times is the design repeated? What are the points of symmetry? What happens when you fold another piece of paper a second time and cut a pattern? How many times is it repeated? What do you think will happen if you fold the paper three times? Four?

U

UNPLUGGED

Variables in Envelopes

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

Variables allow for a lot of freedom in computer science. This lesson helps to explain what variables are and how we can use them in many different ways. Use this activity before (or in conjunction with) the lesson on abstraction to really hit the idea home.

LESSON OBJECTIVES

Students will:

- 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
- Utilize teamwork to enrich creative game play

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Blank Paper
- 3 envelopes per group that have been labeled: robotName, numUnitsTall, & purpose
- 1 Envelope Variables Worksheet per group

- Pens/Pencils/Markers
- Variables Assessment Worksheet

For the Teacher

- 6 or more blank envelopes for warm-up
- Print one Envelope Variables Worksheet per student
- Print one Variables Assessment Worksheet
- Provide students with envelopes (details above), paper, pens & pencils

GETTING STARTED (10 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

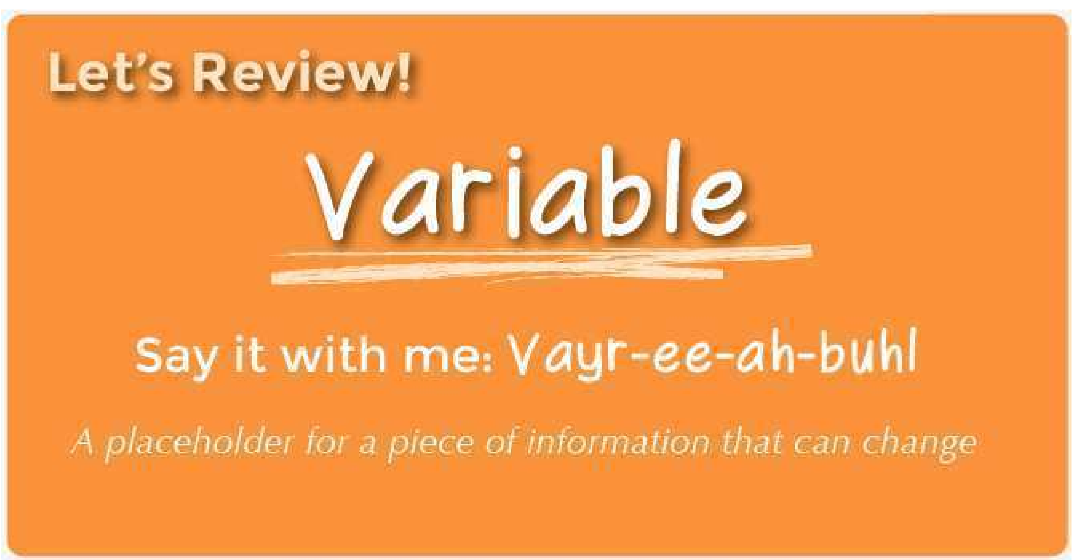
Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we would have had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson??

LESSON TIP *Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.*

2) Vocabulary

This lesson has one important word to review:



Variable - Say it with me: Vayr-ee-ah-buhl
A placeholder for a piece of information that can change

3) Introducing Variables

Call 4 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 them. Maybe 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 you 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.

ACTIVITY: ENVELOPE VARIABLES (20 MIN)

4) Envelope Variables

Once the students understand how the envelopes relate to the sentences, pass out the Robot Variables activity and let them prepare some variables of their own.

Directions:

1) Divide students into groups of 2-4.

2) Have students design (draw) a robot.

3) After 10-15 min, request that the students fill their envelopes with important details about their robot.

4) Collect each group's envelopes, then bring them to the front of the room to share

5) Write on the board, "My robot's name is *robotName*, it is *numUnitsTall* tall, and it's purpose is *purpose*."

6) 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)

5) 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?

6) Vocab Shmocab

You can choose to do these as a class, or have the students discuss with an elbow partner and share.

- Do you remember the definition of the word "variable"?

  "A four sided parallelogram"
  "A placeholder for a piece of information that can change"
  "The wheels on the bottom of chair legs"

ASSESSMENT (10 MIN)

7) Variables Assessment Worksheet

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.

CONNECTIONS AND BACKGROUND INFORMATION

ISTE Standards (formerly NETS) Satisfied by this Lesson Include:

- 1c. Use models and simulations to explore complex systems and issues
- 2d. Contribute to project teams to produce original works or solve problems

- 4b. Plan and manage activities to develop a solution or complete a project
- 6c. Troubleshoot systems and applications

The activities in this lesson support CSTA K-12 Computer Science Standards:

- CL.L2-03 Collaborate with peers, experts, and others using collaborative practices such as pair programming, working in project teams, and participating in group active learning activities
- CT.L1:6-01 Understand and use the basic steps in algorithmic problem-solving
- CT.L1:6-02 Develop a simple understanding of an algorithm using computer-free exercises
- CPP.L1:6-05 Construct a program as a set of step-by-step instructions to be acted out

Next-Gen Science Standards

- 3-5-ETS1-1 Define a simple design problem reflecting a need or a want that includes specified criteria for success and constraints on materials, time, or cost

Common Core Mathematical Practices

- 2. Reason abstractly and quantitatively
- 6. Attend to precision
- 7. Look for and make use of structure
- 8. Look for and express regularity in repeated reasoning

Common Core Language Arts Standards

- L.3.6 Acquire and use accurately grade-appropriate conversational, general academic, and domain-specific words and phrases, including those that signal spatial and temporal relationships
- L.4.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal precise actions, emotions, or states of being and that are basic to a particular topic
- L.5.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal contrast, addition, and other logical relationships

Name: ____________________ Date: ______________

U

Unplugged

Variables in Envelopes

Robot Variables Worksheet

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.

robotName | numUnitsTall | purpose

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

$\boxed{}$ = ________ - ________
magicNumberA = numThree - numOne

$\boxed{}$ = ________ * ________
magicNumberB = numTwo * numOne

$\boxed{}$ = ________ + ________ * ________
magicNumberC = numOne + numTwo * magicNumberB

U

UNPLUGGED

Abstraction with Mad Glibs

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

Abstraction is one of the most important skills for a computer scientist to understand. It simplifies problems and prevents unnecessary repetition. A good coder uses abstraction just about every time she creates a program. This activity will have your students analyze stories for differences so that they can abstract them away. Those abstracted stories become templates for fun and crazy new ones.

TEACHING SUMMARY

Getting Started - 10 minutes

1) Review
2) Vocabulary
3) An Abstract Reminder

Activity: Mad Glibs - 20 minutes

4) Mad Glibs

Wrap-up - 10 minutes

5) Flash Chat: What did we learn?
6) Vocab-Shmocab

Assessment - 10 minutes

7) Abstraction Assessment

LESSON OBJECTIVES

Students will:

- Have the chance to internalize the idea of "abstraction"
- Combine writing and abstraction to test their own creativity
- Analyze their day to find differences that they can turn into similarities

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Mad Glibs Abstraction Worksheet
- Abstraction Assessment Worksheet
- Pens/Pencils

For the Teacher

- Print one Mad Glibs Abstraction Worksheet per student
- Print one Abstraction Assessment Worksheet
- Provide student with paper, pens & pencils

GETTING STARTED (10 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

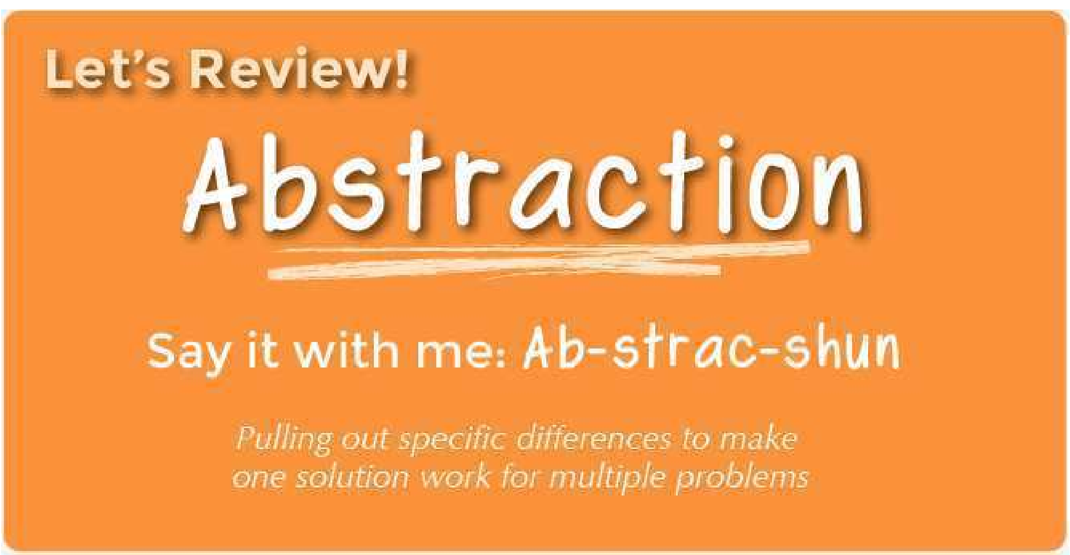
Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we would have had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson??

LESSON TIP *Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.*

2) Vocabulary

This lesson has one important word to review:



Abstraction - Say it with me: Ab-strac-shun
Pulling out specific differences to make one solution work for multiple problems

3) An Abstract Reminder

When you finish your review, try taking your class by surprise.

> "So, what did you have for waffles this morning?"

Your students might look perplexed?

"No one? Okay, what did you have for toast yesterday?"

You may start to get some hands raising, and people eager to share and agree because they want to relate, but possibly not because they understand.

"See what I was doing there? I identified my experience in a very specific manner, and that made it harder for everyone else to relate to. What could I have said that more people would have understood?"

At some point, they'll start to come up with the idea of using "breakfast" in place of the actual food that was consumed.

"In a way, the word 'breakfast' is like a variable that we use to hold a space for whatever it is we ate this morning. By taking the specific word out and replacing the space it leaves with 'breakfast,' we are using abstraction to make something work for multiple people."

Ask the class to give you some examples of other places that they may naturally use abstraction to allow more people to understand them. The idea of lunch and dinner will most-likely come up. Is there anything *not* food related?

ACTIVITY: MAD GLIBS (20 MIN)

4) Mad Glibs Abstraction Worksheet

The next step is to pass out a "fill-in-the-blank" story (see the Mad Glibs Abstraction Worksheet for a copy to print out) . Let them know that this started as a specific story about one thing, but we used abstraction to turn some of the specific words into blanks, and now the story can be about lots of things. Ask them what they can make their story about.

Story 1

First you take your ________________ then add a layer of ________________

before you pour on a hearty dose of ________________ .

Next, press some ________________ down into the ________________ before

covering with a sprinkle of ________________ .

That's how I make a ________________ !

WRAP-UP (10 MIN)

5) Flash Chat: What did we learn?

- What did we learn today?
- How might you be able to use abstraction when describing things outside?
- Can you think of ways to use abstraction when talking about animals?

6) Vocab Shmocab

You can choose to do these as a class, or have the students discuss with an elbow partner and share.

- Do you remember the definition of the word "abstraction"?

"A list of steps that you can follow to finish a task"
"An object that looks like a triangle from one direction and a circle from another"
"Pulling out specific differences to make one solution work for multiple problems"

ASSESSMENT (10 MIN)

7) [Abstraction Assessment Worksheet]

EXTENDED LEARNING

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

Mad Drawing

- Challenge the students to use these same ideas to make puzzles out of drawings.
  - Have the students fold two pieces of paper in half three times and re-open them to lay them flat.

  - On one of those pieces, have them make a drawing where they leave one of the resulting rectangles blank.
  - Have them cut the other folded piece along the creases to wind up with eight rectangle shapes.
    - Encourage the students to make different drawings on all eight rectangles that believably complete their original drawing.
- Be sure to allow students time to share.

Prep for Songwriting with Parameters

- Can students find songs where there are phrases that repeat, except for one or two words that change?
  - Using "Old MacDonald" or "Five Little Monkeys" will prepare the students nicely for an activity that is coming up.

CONNECTIONS AND BACKGROUND INFORMATION

ISTE Standards (formerly NETS) Satisfied by this Lesson Include:

- 1a. Apply exitisng knowledge to generate new ideas, products, or processes
- 1c. Use models and simulations to explore complex systems and issues
- 2d. Contribute to project teams to produce original works or solve problems
- 4b. Plan and manage activities to develop a solution or complete a project

The activities in this lesson support CSTA K-12 Computer Science Standards:

- CL.L2-03 Collaborate with peers, experts, and others using collaborative practices such as pair programming, working in project teams, and participating in group active learning activities
- CT.L1:6-01 Understand and use the basic steps in algorithmic problem-solving
- CT.L1:6-02 Develop a simple understanding of an algorithm using computer-free exercises
- CT.L2-12 Use abstraction to decompose a problem into sub problems.

Common Core Mathematical Practices

- 2. Reason abstractly and quantitatively.
- 6. Attend to precision.
- 7. Look for and make use of structure.
- 8. Look for and express regularity in repeated reasoning.

Common Core Language Arts Standards

- L.3.6 Acquire and use accurately grade-appropriate conversational, general academic, and domain-specific words and phrases, including those that signal spatial and temporal relationships.
- L.4.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal precise actions, emotions, or states of being and that are basic to a particular topic.
- L.5.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal contrast, addition, and other logical relationships

Name: ____________________ Date: ______________

U

Unplugged

Mad Glibs

Abstraction Worksheet

Write a story using the Mad Glibs template below. Fill in the blanks with words to create something fun to share. Then, create a second story by writing another version on the lines at the bottom of the page.

Story 1

First you take your ______________ then add a layer of ______________

before you pour on a hearty dose of ______________ .

Next, press some ______________ down into the ______________ before

covering with a sprinkle of ______________ .

That's how we make a ____________________ !

Story 2

____

____

____

____

____

____

____

____

____

Name: ______________________ Date: ______________

U

Unplugged

Mad Glibs

Abstraction Assessment Worksheet

CODE

The Mad Glib template that we used to make these stories has vanished! Look at the stories and figure out which words are supposed to be blanks, then recreate the template at the bottom of the page.

Story 1

Early last year, my mom gave me an old skateboard. She told me about the days when she would ride it from her school in her hometown. I tried to ride it once, but tripped over my shoelaces. It didn't take long before I decided that it was best to leave the skateboarding to my mom.

Story 2

Sometime last year, my mom told me an old story. She told me about the days when she would hear it from her father in her childhood. I tried to tell it once, but tripped over my words. It didn't take long before I decided that it was best to leave the storytelling to my mom.

Create new template here:

6

Course 4 | Lesson 6

Artist: Variables

Lesson time: 30 Minutes

LESSON OVERVIEW

In this lesson, students will explore the creation of repetitive designs using variables. Students will learn how variables can be used to make code more simple to write and easier to read.

LESSON OBJECTIVES

Students will:

- Create programs that utilize repetition to create gorgeous designs
- Use trial and error to recreate detailed designs in proper scale
- Calculate angles by dividing 360 by the number of sides in a polygon
- Decompose a shape into its smallest repeatable sequence

GETTING STARTED

Introduction

Remind your students of the unplugged lesson from a previous class.

- What if you wanted to draw a square on the board and each side was labeled "side"
  - What would happen if you had an envelope labeled "side" with "10 inches inside"?
  - What would happen to the square if you switched the paper in the "side" envelope to "20 inches"?
  - What if you labeled the sides of the square with "2*side"?

Next, review with students the basic artist navigation, particularly:

- Moving forward
- Turning left/right
- Looping
- Angles

ACTIVITY

Artist: Variables

This lesson explores the use of variables as a way to quickly change many values at one time. Not only will the students be dealing with the looping of designs and repetition of angles, they'll also be doing math on variables. It can be helpful for them to have paper and pencil to figure out values as they go. Also, let them know ahead of time that there will likely be some puzzles that confuse or frustrate them. This is normal, and expected. Students should prepare themselves for persistence and perseverance.

EXTENDED LEARNING

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

Get Steppin'

Get a few volunteers to come to the front of the room. Assign the group a shape (like a line, a triangle, or a square) with each side length "side". Give each volunteer an envelope labeled "side" with different numbers of steps inside of each. Have them all start walking to make whatever shape you assigned, but each only gets to walk the number of steps inside their envelope before they turn.

- Do one sample, then ask the class if they can tell where the variable came in
- After you've done the square, ask the class how we might be able to use the same variable to create a rectangle with a length that's twice the width.
- How else might we use the same variable to change our polygons?
- Where could we use a second variable? What might we call it?

7

Course 4 | Lesson 7

Play Lab: Variables

Lesson time: 30 Minutes

LESSON OVERVIEW

In this activity, students will have the opportunity to play with variables in a situation that illustrates just how useful they can be. Students will edit games to give themselves the advantage and make their characters more powerful using variables as parameters.

LESSON OBJECTIVES

Students will:

- Identify the numbers that are responsible for specific elements of a program
- Create a game that incorporates numerical parameters
- Replace numbers with descriptive variables

GETTING STARTED

Introduction

Review the previous lesson, paying particular attention to the use of variables.

- What is a variable, and how many ways can you think to use them?
- Now you're going to create games online using variables instead of entering numbers.

ACTIVITY

Play Lab: Variables

This lesson will guide the students from a place where they are playing a game programmed using numbers traditionally, to a place where they substitute variables for numerical values so that their program is easier to read.

The challenges with this stage come about most when they are trying to remember to use variables in free play at the end. It may be helpful to walk around and ask the students to show you where and how they are using variables, and why they chose the names that they came up with.

EXTENDED LEARNING

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

Variable Surprise

Bring your students back to Play Lab and have them create any game they want, with the requirement that they each have variables called "step" and "fly".

- Once students have had a chance to make something, encourage the class to walk around to look at one another's programs.
  - Did any of the games have similarities?
  - How might the variable names have influenced the creation of the games?
- Have the students go back to edit their games.
  - Ask the students to set "step" to 2 and "fly" to 20 and share out how that changed their original creations.
    - Did it affect anyone in a way that was unexpected?
    - Note that the students all had the same variable names, but they likely used them differently.

U

UNPLUGGED

For Loop Fun

Lesson time: 25 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON 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 with extra structure built right in.

TEACHING SUMMARY

Getting Started - 15 minutes

1) Review
2) Vocabulary
3) For One and All

Activity: For Loop Fun - 25 minutes

4) For Loop Fun

Wrap-up - 10 minutes

5) Flash Chat - What did we learn?
6) Vocab-Shmocab

Assessment - 5 minutes

7) Conditionals with Cards Assessment

LESSON OBJECTIVES

Students will:

- Determine starting value, stopping value, and interval of "for loop"
- Illustrate the counter values hit each time through a for loop during runtime

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Dice (3 per pair of students)
- Pens & Pencils
- For Loop Fun Worksheet
- For Loop Fun Assessment

For the Teacher

- This Teacher Lesson Guide
- Print one For Loop Fun Worksheet per group
- Print one For Loop Fun Assessment for each student

GETTING STARTED (20 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

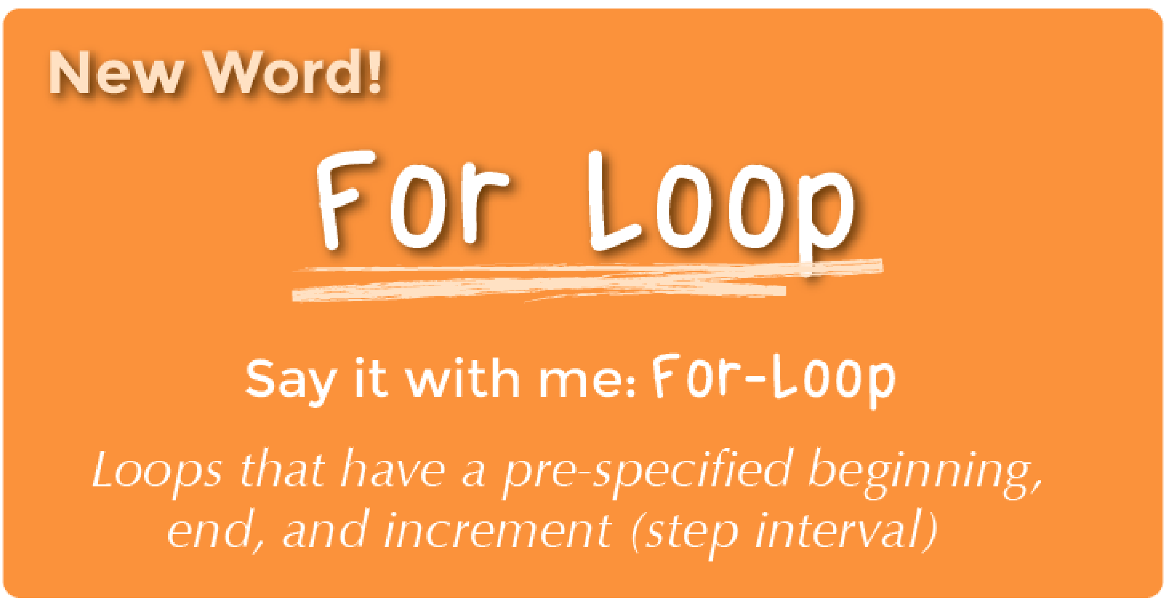
Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we would have had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson??

LESSON TIP *Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.*

2) Vocabulary

This lesson has one new and important word:



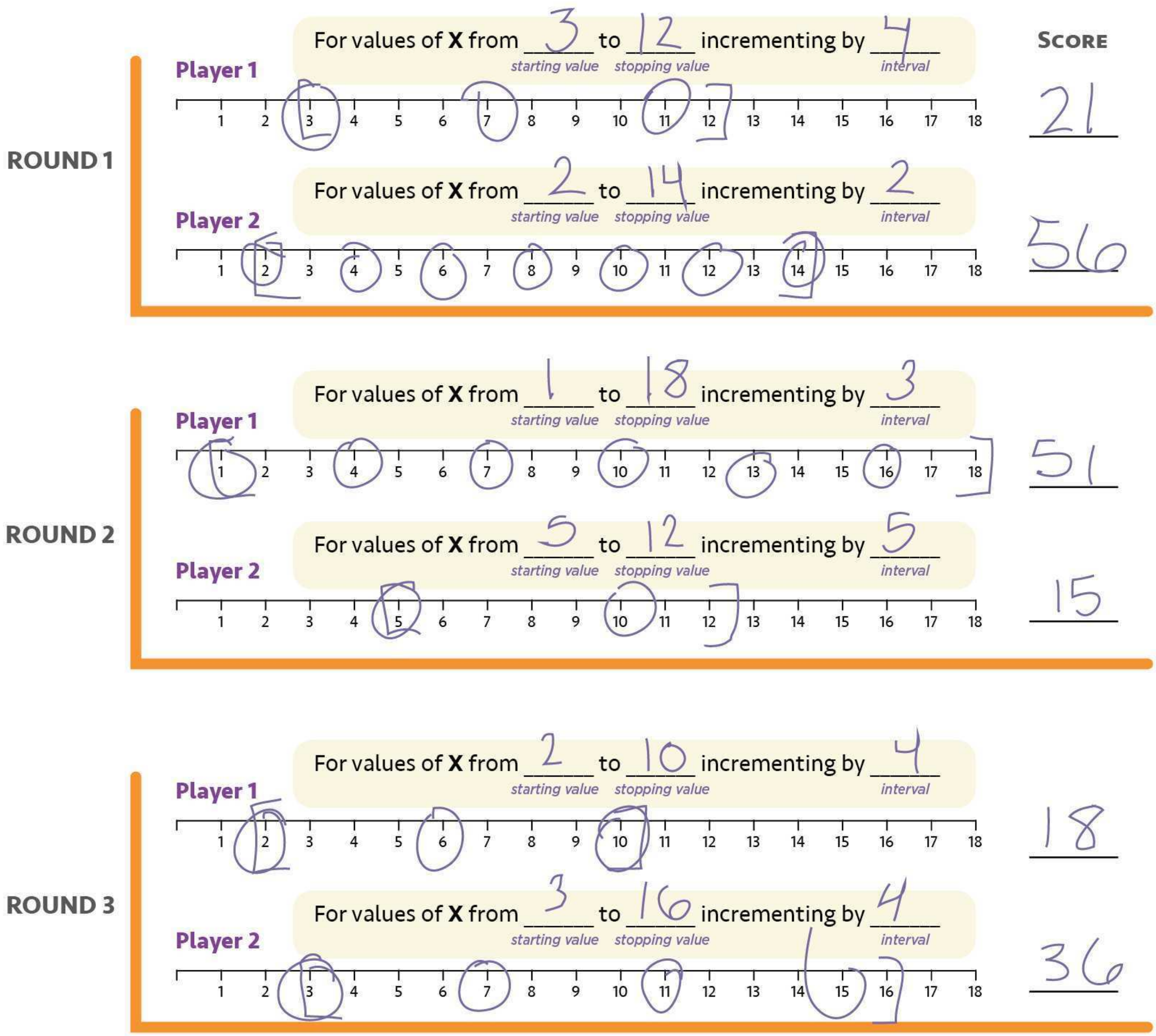
For Loop - Say it with me: For-Loop
Loops that have a predetermined beginning, end, and increment (step interval)

3) For One and All

- If you did the original loops lesson (remember The Iteration?) you can call back to the usefulness of loops in general
- 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

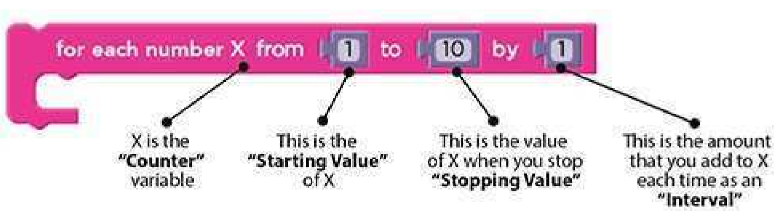
Here, you can jump right into a sample of the game



ACTIVITIES: (20 MIN)

4) For Loop Fun

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.



Directions:

1) Divide students into pairs

2) 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

3) 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

4) Add all of the circled values to your score, then let the other player take a turn

5) 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 (5 MIN)

5) 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.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

"Doing something more than once the exact same way"
"Loops that have a predetermined beginning, end, and increment."
"Statements that only run under certain conditions"

...and what is the word that we learned?

ASSESSMENT (5 MIN)

7) 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

CONNECTIONS AND BACKGROUND INFORMATION

ISTE Standards (formerly NETS) Satisfied by this Lesson Include:

- 1c. Use models and simulation to explore complex systems and issues
- 2d. Contribute to project teams to solve problems

The activities in this lesson support CSTA K-12 Computer Science Standards:

- CL.L1:3-02 Work cooperatively and collaboratively with peers teachers, and others using technology
- CT.L1:6-01 Understand and use the basic steps in algorithmic problem-solving
- CT.L1:6-02 Develop a simple understanding of an algorithm using computer-free exercises
- CT.L2-01 Use the basic steps in algorithmic problem solving to design solutions
- CT.L2-12 Use abstraction to decompose a problem into sub-problems
- CT.L2-14 Examine connections between elements of mathematics and computer science including binary numbers, logic, sets, and functions
- CT.L3A-03 Explain how sequence, selection, iteration, and recursion are building blocks of algorithms

Next-Gen Science Standards

- 3-5-ETS1-2 Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem

Common Core Mathematical Practices

- 1. Make sense of problems and persevere in solving them
- 2. Reason abstractly and quantitatively.
- 4. Model with mathematics
- 6. Attend to precision
- 7. Look for and make use of structure
- 8. Look for and express regularity in repeated reasoning

Common Core Math Standards

- 4.OA.C.5 Generate a number or shape pattern that follows a given rule. Identify apparent features of the pattern that were not explicit in the rule itself

Common Core Language Arts Standards

- L.3.6 Acquire and use accurately grade-appropriate conversational, general academic, and domain-specific words and phrases, including those that signal spatial and temporal relationships
- L.4.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal precise actions, emotions, or states of being and that are basic to a

particular topic

- L.5.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal contrast, addition, and other logical relationships

Name: ____________________ Date: ______________

U

Unplugged

For Loop Fun

Number Lines and Score Sheet

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

Score

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

Score: ______

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

Score: ______

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 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

Score: ______

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: ______

Name: ____________ Date: ____________

U

Unplugged

For Loop Fun

Sample Game Sheet

CODE

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

SCORE

ROUND 1

Player 1

For values of **X** from 3 to 12 incrementing by 4
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

21

Player 2

For values of **X** from 2 to 14 incrementing by 2
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

56

ROUND 2

Player 1

For values of **X** from 1 to 18 incrementing by 3
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

51

Player 2

For values of **X** from 5 to 12 incrementing by 5
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

15

ROUND 3

Player 1

For values of **X** from 2 to 10 incrementing by 4
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

18

Player 2

For values of **X** from 3 to 16 incrementing by 4
starting value *stopping value* *interval*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

36

Name: ______________________ Date: ______________

Unplugged

For Loop Fun

Assessment Worksheet

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?

Score

ROUND 1

Player 1

For values of **X** from 1 (starting value) to 18 (stopping value) incrementing by 4 (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 3 (starting value) to 11 (stopping value) incrementing by 2 (interval)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

______

ROUND 2

Player 1

For values of **X** from 3 (starting value) to 17 (stopping value) incrementing by 5 (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 5 (starting value) to 17 (stopping value) incrementing by 3 (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 6 (starting value) to 11 (stopping value) incrementing by 1 (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 2 (starting value) to 15 (stopping value) incrementing by 6 (interval)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

______

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

9

Course 4 | Lesson 9

Bee: For Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Students use the Bee environment to write programs that use loops with embedded counters/index variables. These loops are called for loops, and they utilize predetermined start and stop values.

LESSON OBJECTIVES

Students will:

- Break one long sequence of steps into shorter looped sequences
- Use the "for loop" structure to repeat an action a variable number of times each iteration

GETTING STARTED

Introduction

ACTIVITY

Bee: For Loops

This activity walks students through the differences between loops and for loops. In many cases, it is possible to solve problems with brute force that can easily be solved using for loops instead. As you walk around, look for students who are correctly using for loops and counters, then point out why the counters make the program easier.

The final stage counts down, collecting less nectar as the counter increases. This is expected to cause a small amount of frustration. Encourage students to talk about what is supposed to be happening out loud. That should help them stumble upon the answer on their own.

10

Course 4 | Lesson 10

Artist: For Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Students use the Artist environment to write programs with for loops, similar to what they did in the previous Bee level.

TEACHING SUMMARY

Getting Started

Introduction

Activity: Artist: For Loops

Artist: For Loops

Extended Learning

Extension Activities

LESSON OBJECTIVES

Students will:

- Predict the number of steps needed to increment in each for loop iteration
- Determine start and stop values for multiple for loop examples

GETTING STARTED

Introduction

ACTIVITY

Artist: For Loops

EXTENDED LEARNING

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

11

Course 4 | Lesson 11

Play Lab: For Loops

Lesson time: 30 Minutes

LESSON OVERVIEW

Building on the previous Play Lab activity, students will add deeper interactivity as they build their own video games.

TEACHING SUMMARY

LESSON OBJECTIVES

Students will:

- Utilize for loops to count from 1 to 100
- Count by tens repeatedly using the for loop structure
- Employ skills from previous lessons to create more difficult looping algorithms

GETTING STARTED

Introduction

ACTIVITY

Play Lab: For Loops

Play Lab allows students to combine their new for loop skills with skills from past lessons to create fun games that are relevant to their life. The last puzzle of this stage is fairly unstructured, so a watchful eye may be helpful to encourage the class to integrate for loops into their final game.

EXTENDED LEARNING

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

12

Course 4 | Lesson 12

Artist: Functions

Lesson time: 30 Minutes

LESSON OVERVIEW

Students use the Artist environment to draw complicated images using functions for repeated tasks.

TEACHING SUMMARY

Getting Started

Introduction

Activity: Artist: Functions

Artist: Functions

Extended Learning

Extension Activities

LESSON OBJECTIVES

Students will:

- Identify repeated movements and utilize functions to simplify their program
- Use trial and error to re-create complex patterns
- Break complex tasks into smaller repeatable sections
- Combine simple shapes into complex designs with functions

GETTING STARTED

Introduction

In this stage, students will walk from using repetitive instructions into learning how to integrate functions as a simplification technique.

ACTIVITY

Artist: Functions

EXTENDED LEARNING

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

U

UNPLUGGED

Songwriting with Parameters

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON 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.

LESSON OBJECTIVES

Students will:

- Locate repeating phrases inside song lyrics
- Identify sections of a song to pull into a function (chorus)
- Modify functions to accept parameters
- Describe how functions and parameters can make programs easier to write

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- Songwriting Worksheets
- Pens & Pencils
- One Songwriting Assessment for each student.

For the Teacher

- This Teacher Lesson Guide
- Print several Songwriting Worksheets for each group
- Print one Songwriting Assessment for each student.
- Access to the internet, or pre-downloaded songs and lyrics for activity

GETTING STARTED (20 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

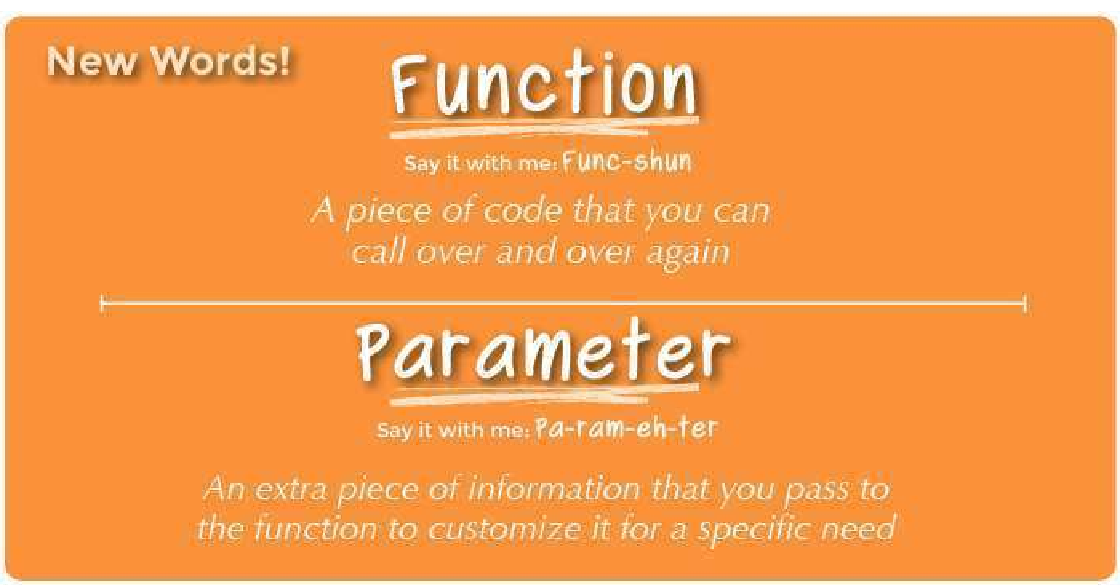
Here are some questions that you can ask in review:

- What did we do last time?
- What do you wish we would have had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson?

LESSON TIP *Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.*

2) 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

3) 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 more 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 the some of the same reasons that it is helpful to songwriters. - 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!

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.

ACTIVITIES: (20 MIN)

4) Songwriting

- 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

- 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.

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.

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

WRAP-UP (5 MIN)

5) 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.

6) Vocab Shmocab

- Which one of these definitions did we learn a word for today?

  "A piece of code that you can call over and over again"
  "A small shard of wood"
  "Getting help from a large group of people to finish something faster"

  ...and what is the word that we learned?

- What about parameters?
  - Why do we use them?
  - Can you pass more than one parameter to a chorus function?
  - Can you pass more than two?

ASSESSMENT (5 MIN)

7) 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.

- 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?

CONNECTIONS AND BACKGROUND INFORMATION

ISTE Standards (formerly NETS)

- 1a. Apply existing knowledge to generate new ideas, products, or processes
- 1c. Use models and simulation to explore complex systems and issues
- 2a. Interact, collaborate, and publish with peers, experts, or others employing a variety of digital environments and media
- 2d. Contribute to project teams to solve problems
- 4b. Plan and manage activities to develop a solution or complete a project
- 4d. Use multiple processes and diverse perspectives to explore alternative solutions

CSTA K-12 Computer Science Standards

- CL.L1:3-02 Work cooperatively and collaboratively with peers teachers, and others using technology
- CT.L2-01 Use the basic steps in algorithmic problem solving to design solutions
- CT.L2-06 Describe and analyze a sequence of instructions being followed
- CT.L2-07 Represent data in a variety of ways: text, sounds, pictures, numbers
- CT.L2-08 Use visual representations of problem states, structures, and data
- CT.L2-12 Use abstraction to decompose a problem into sub problems
- CT.L2-14 Examine connections between elements of mathematics and computer science including binary numbers, logic, sets, and functions
- CT.L3A-01 Use predefined functions and parameters, classes and methods to divide a complex problem into simpler parts
- CT.L3A-03 Explain how sequence, selection, iteration, and recursion are building blocks of algorithms
- CPP.L1:6-05 Construct a program as a set of step-by-step instructions to be acted out

NGSS Science and Engineering Practices

- 3-5-ETS1-2 Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem

Common Core Mathematical Practices

- 1. Make sense of problems and persevere in solving them
- 2. Reason abstractly and quantitatively
- 3. Construct viable arguments and critique the reasoning of others
- 6. Attend to precision
- 7. Look for and make use of structure
- 8. Look for and express regularity in repeated reasoning

Common Core Language Arts Standards

- SL.3.1.D Explain their own ideas and understanding in light of the discussion
- SL.3.3 Ask and answer questions about information from a speaker, offering appropriate elaboration and detail
- RI.3.1 Ask and answer questions to demonstrate understanding of a text, referring explicitly to the text as the basis for the answers
- L.3.6 Acquire and use accurately grade-appropriate conversational, general academic, and domain-specific

words and phrases, including those that signal spatial and temporal relationships

- L.4.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal precise actions, emotions, or states of being and that are basic to a particular topic
- L.5.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal contrast, addition, and other logical relationships

Group Name: ______________________ Date: ______________

Unplugged

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-iiiiiiiiiii
eeeeeeee-iiiiiiiiiii
ohhhhhhhhhhhh!

Unplugged

Group Name: ______________________ Date: ______________

Songwriting Worksheet

Using Lyrics to Explain Functions and Procedures

CODE

Song Name:

Chorus:

Parameter Examples: ______________ (P1) ______________ (P2) ______________ (P3)

Song:

Name: ____________________ Date: ______________

U

Unplugged

Songwriting

Lesson Assessment

CODE

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.

Name: ______________________ Date: ______________

U

Unplugged

Songwriting Worksheet

Lesson 8 Assessment - Finding the Function in a Song

Song Name:

Chorus:

Parameter Examples: ______________ (P1) ______________ (P2) ______________ (P3)

Song:

14

Course 4 | Lesson 14

Artist: Functions with Parameters

Lesson time: 30 Minutes

LESSON OVERVIEW

Students will use the Artist environment to draw complicated images using functions with parameters to create similar shapes with small differences.

LESSON OBJECTIVES

Students will:

- Identify repeated movements and utilize functions to simplify a program
- Break complex tasks into smaller repeatable sections
- Combine simple shapes into complex designs with functions
- Utilize parameters to make one function work for multiple purposes

GETTING STARTED

Introduction

Students will have experienced functions through Artist in a previous stage. This stage adds the option of parameters, which will allow one function to work for multiple creations. There may be some confusion over creating parameters inside a function, so it is a good idea to have the class watch the associated video, or to watch a function with parameter being created in advance of the lesson.

ACTIVITY

Artist: Functions with Parameters

EXTENDED LEARNING

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

enrichment.

15

Course 4 | Lesson 15

Play Lab: Functions with Parameters

Lesson time: 30 Minutes

LESSON OVERVIEW

Having experienced the creation and use of functions and parameters, students will get the opportunity to use the skill in the creation of Play Lab games. Later puzzles incorporate the use of multiple parameters.

LESSON OBJECTIVES

Students will:

- Identify repeated movements and utilize functions to simplify a program
- Utilize parameters to make one function work for multiple purposes
- Adapt their understanding of functions to allow for the use of multiple parameters

GETTING STARTED

Introduction

While the students will have had some experience creating functions with parameters in earlier stages, this Play Lab stage adds the complexity of using multiple parameters inside a single function. This is a great opportunity to remind students that it is okay to get frustrated and to use trial and error as an effective learning tool.

ACTIVITY

Play Lab: Functions with Parameters

EXTENDED LEARNING

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

16

Course 4 | Lesson 16

Bee: Functions with Parameters

Lesson time: 30 Minutes

LESSON OVERVIEW

This short stage illustrates how students can use their new skills with functions and parameters to change direction using binary logic.

LESSON OBJECTIVES

Students will:

- Edit existing functions to make them work for specific tasks
- Combine similar functions into a single one by utilizing parameters

GETTING STARTED

Introduction

This lesson gets complicated rather quickly. If your students had an easy time with previous lessons, then they should be able to keep up just fine. If, however, they found the previous stages difficult, you may want to have them revisit Play Lab: Functions with Paramaters first. In this Bee level, students will not only be using multiple parameters iside their functions, they will also be using binary flags (left = 0, right = 1) to allow for logic decisions.

ACTIVITY

Bee: Functions with Parameters

EXTENDED LEARNING

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

U

UNPLUGGED

Binary Images

Lesson time: 20 Minutes Basic lesson time includes activity only. Introductory and Wrap-Up suggestions can be used to delve deeper when time allows.

LESSON OVERVIEW

Though many people think of binary as strictly zeros and ones, our previous courses taught students 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.

TEACHING SUMMARY

Getting Started - 10 minutes

1) Review
2) Vocabulary
3) Binary in Review

Activity: Binary Images - 20 minutes

4) Binary Images

Wrap-up - 10 minutes

5) Flash Chat: What did we learn?
6) Vocab-Shmocab

Assessment - 10 minutes

7) Binary Images Assessment

LESSON OBJECTIVES

Students will:

- Identify methods for encoding images into binary
- Relate images to a peer using binary encoding
- Reproduce an image, based on binary code

TEACHING GUIDE

MATERIALS, RESOURCES AND PREP

For the Student

- One Binary Images Worksheet per pair
- One Binary Image Assessment Worksheet per student
- Blank Paper, pens, pencils

- Other trinkets that can display opposites, such as: Playing Cards, checkers, coin, etc. (Optional)

For the Teacher

- Teacher guide
- Print one Binary Images Worksheet per pair
- Print one Binary Image Assessment Worksheet per student
- Gather groupings of items that can show opposites for students to use when coming up with their own binary encodings (Optional)

GETTING STARTED (10 MIN)

1) Review

This is a great time to review the last lesson that you went through with your class. We suggest you alternate between asking questions of the whole class and having students talk about their answers in small groups.

Here are some questions that you can ask in review:

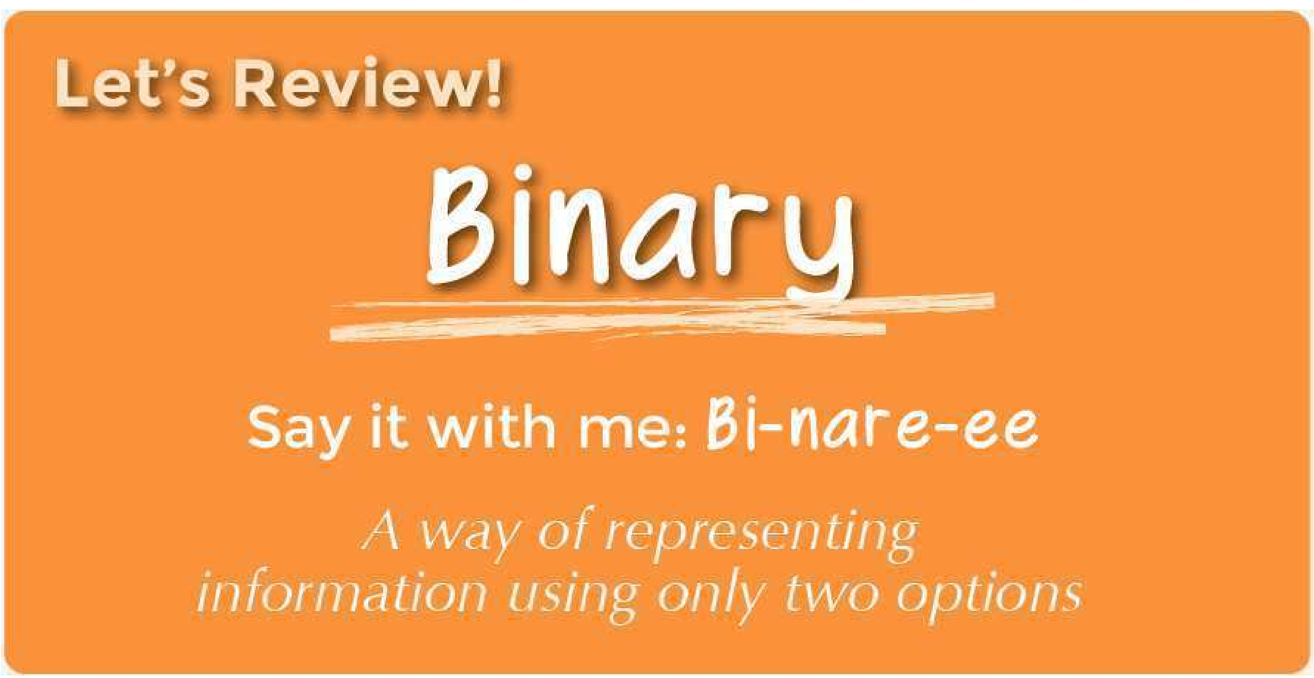
- What did we do last time?
- What do you wish we would have had a chance to do?
- Did you think of any questions after the lesson that you want to ask?
- What was your favorite part of the last lesson??

LESSON TIP

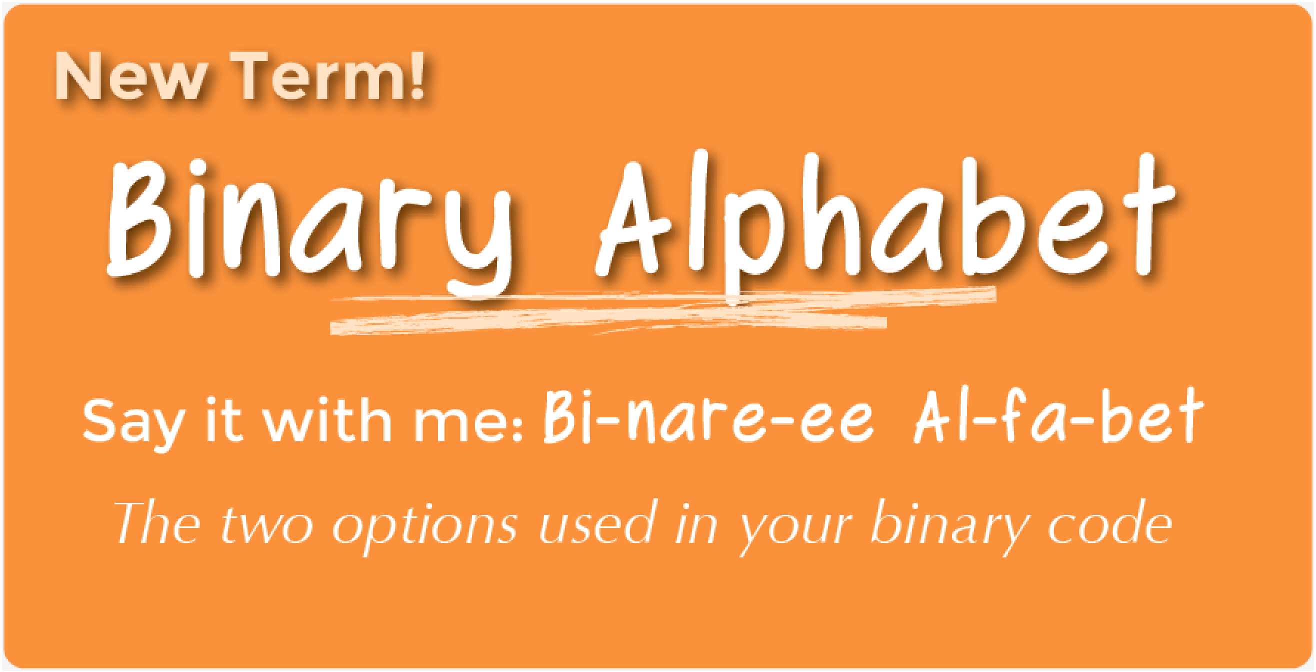
Finishing the review by asking about the students' favorite things helps to leave a positive impression of the previous exercise, increasing excitement for the activity that you are about to introduce.

2) Vocabulary

This lesson has one important word to review and one new term:



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

3) Binary in Review

See if your class remembers the [Binary Bracelets] lesson from course 2.

> "Do you remember how we used off and on to represent letters?"

You may want to do an example with them using a letter or two to refresh their memory.

> "What other ways could we have represented those letters? What if we couldn't use on and off?"

Encourage your class to come up with some other sets of opposites to represent the same letters that you just did.

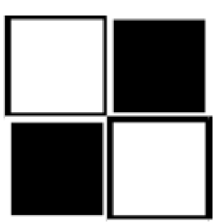
> "Those are great suggestions. Let's use one of them to code a new letter."

Once you code up another letter or two, begin a though exercise.

> "How could we use that same binary alphabet to encode a picture?"

The students may have no idea what you're talking about. That's okay. You can lead them to the answer step-by step.

What if we had a picture like this, where there's only two different 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 harken 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!"

ACTIVITY: BINARY IMAGES (20 MIN)

4) Binary Images

Now it's the students' turn!

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.

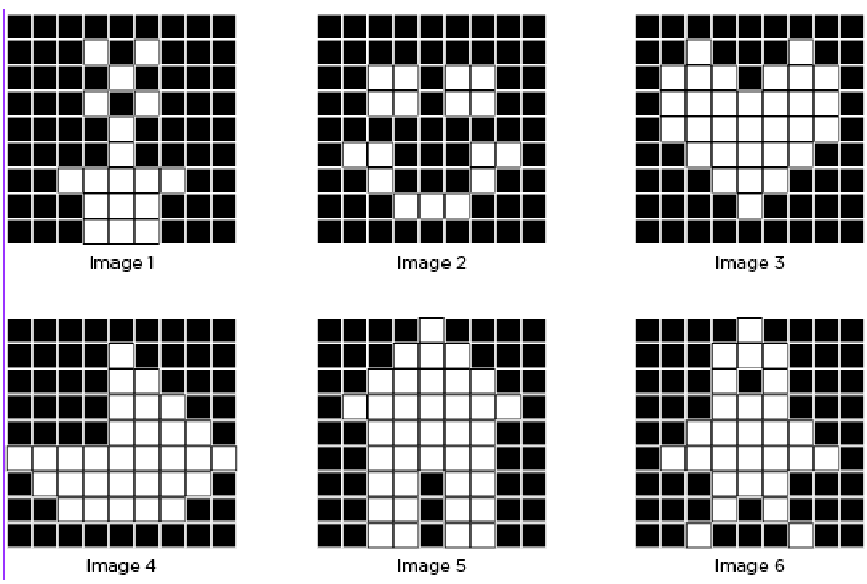


Image 1 Image 2 Image 3

Image 4 Image 5 Image 6

WRAP-UP (10 MIN)

5) 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?

6) Vocab Shmocab

You can choose to do these as a class, or have the students discuss with an elbow partner and share.

- Do you remember the definition of the term "binary alphabet"?

"The two options used in your binary code"

"A three sided polygon"
"A number larger than zero"

ASSESSMENT (10 MIN)

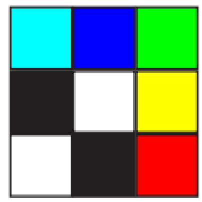
7) [Binary Image Assessment Worksheet]

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.

CONNECTIONS AND BACKGROUND INFORMATION

ISTE Standards (formerly NETS) Satisfied by this Lesson Include:

- 1c. Use models and simulations to explore complex systems and issues
- 2d. Contribute to project teams to produce original works or solve problems
- 4b. Plan and manage activities to develop a solution or complete a project
- 4d. Use multiple processes and diverse perspectives to explore alternative solutions
- 6d. Transfer current knowledge to learning new technologies

The activities in this lesson support CSTA K-12 Computer Science Standards:

- CL.L1:3-02 Work cooperatively and collaboratively with peers teachers, and others using technology
- CT.L1:6-01 Understand and use the basic steps in algorithmic problem-solving
- CL.L2-03 Collaborate with peers, experts, and others using collaborative practices such as pair programming, working in project teams, and participating in group active learning activities
- CT.L2-06 Describe and analyze a sequence of instructions beign followed
- CT.L2-07 Represent data in a variety of ways: text, sounds, pictures, numbers
- CT.L2-14 Examine connections between elements of mathematics and computer science including binary numbers, logic, sets, and functions
- CT.L3A-05 Describe the relationsship between binary and hexademinal representations
- CT.L3B-07 Discus the interpretation of binary sequences in a variety of forms
- CT.L1:6-02 Develop a simple understanding of an algorithm using computer-free exercises

Next-Gen Science Standards

- 3-5-ETS1-2 Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem

Common Core Mathematical Practices

- 1. Make sense of problems and persevere in solving them
- 2. Reason abstractly and quantitatively

- 4. Model with mathematics
- 6. Attend to precision
- 7. Look for and make use of structure
- 8. Look for and express regularity in repeated reasoning

Common Core Math Standards

- 4.OA.C.5 Generate a number or shape pattern that follows a given rule. Identify apparent features of the pattern that were not explicit in the rule itself

Common Core Language Arts Standards

- L.3.6 Acquire and use accurately grade-appropriate conversational, general academic, and domain-specific words and phrases, including those that signal spatial and temporal relationships
- L.4.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal precise actions, emotions, or states of being and that are basic to a particular topic
- L.5.6 Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal contrast, addition, and other logical relationships

Name: ____________________ Date: __________________

U

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.

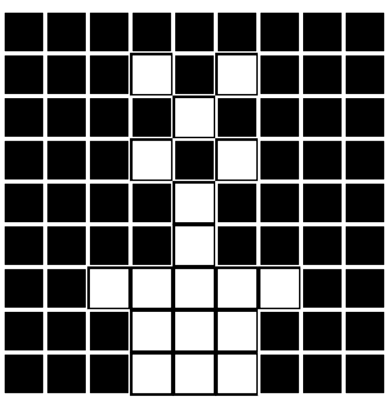


Image 1

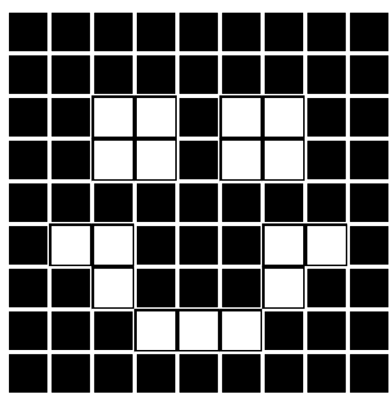


Image 2

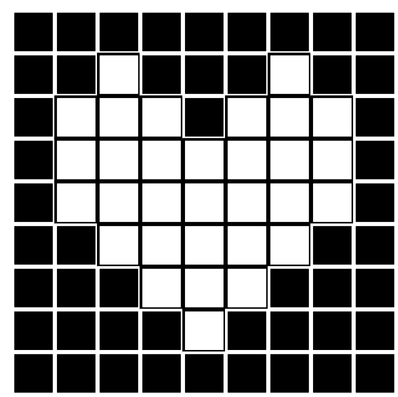


Image 3

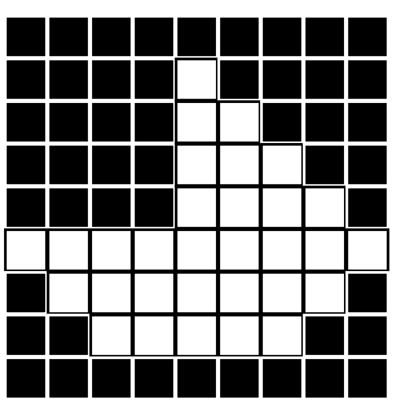


Image 4

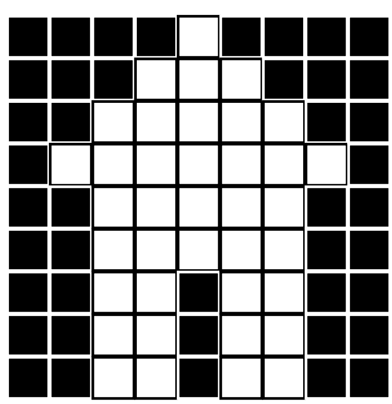


Image 5

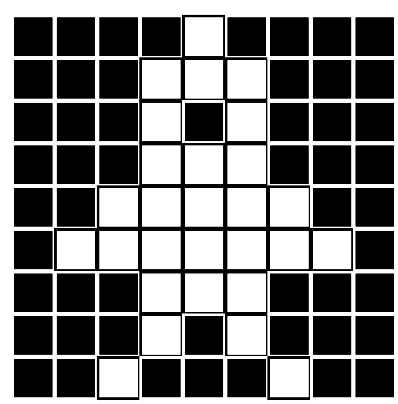


Image 6

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.

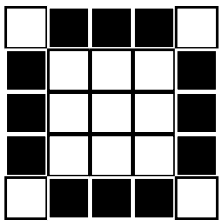


image #1

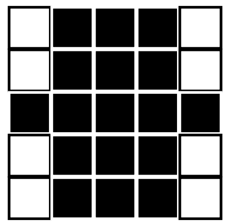


image #2

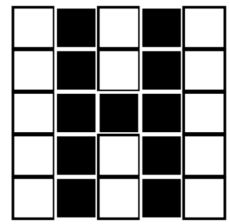


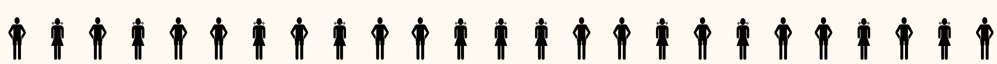
image #3

A) ★✖✖✖★★✖✖✖★✖✖✖✖✖★✖✖✖★★✖✖✖★

✖ = ____________________ ★ = ____________________ This encodes image # __________

B) ○●●●○●○○○●●○○○●●○○○●○●●●○

○ = ____________________ ● = ____________________ This encodes image # __________

C) 

 = ____________________  = ____________________ This encodes image # __________

How do you know that your answers are correct?

____

____

____

____

18

Course 4 | Lesson 18

Artist: Binary

Lesson time: 30 Minutes

LESSON OVERVIEW

Rounding out Course 4 is the Artist: Binary lesson. Here, students will build binary images, translating 0s and 1s to offs and ons (or blacks and whites).

LESSON OBJECTIVES

Students will:

- Match binary sequences to encoded images
- Utilize loops and binary code to recreate provided images
- Identify repeated sequences and break long codes up into smaller chunks that can be looped
- Create pictures using unique combinations of on and off

GETTING STARTED

Introduction

To begin, it can be helpful to review the previous lesson, specifically different ways of using binary to indicate how to create an image on a grid. This stage will translate the unplugged activity into a simple, independent online lesson.

ACTIVITY

Artist: Binary

EXTENDED LEARNING

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

19

Course 4 | Lesson 19

Super Challenge - Variables

Lesson time: 30 Minutes

LESSON OVERVIEW

In this lesson, students will be challenged to dig deep into what they have learned throughout their journey to solve a handful of complex puzzles.

LESSON OBJECTIVES

Students will:

- Create programs that utilize repetition to create gorgeous designs
- Decompose large, difficult puzzles into managable pieces
- Use variables to capture patterns in complex tasks

GETTING STARTED

Introduction

Remind your students of the lessons that they have done with variables previously.

- What kinds of things are variables good for?
  - How does it help to move by "length" amount using artist?
  - Why is it helpful to have the bee grab "nectarUnits" of nectar?
  - Can you use variables as numbers in math equations?
- Next, remind your students that this stage is intended to be very difficult and most puzzles will take several attempts to solve. [/together]

ACTIVITY

Super Challenge - Variables

This lesson explores the use of variables as a way to quickly change many values at one time. Not only will the students be dealing with the looping of designs and repetition of angles, they'll also be doing math on variables. It can be helpful for them to have paper and pencil to figure out values as they go. Also, let them know ahead of time that there will likely be some puzzles that confuse or frustrate them. This is normal, and expected. Students should prepare themselves for persistence and perseverance.

20

Course 4 | Lesson 20

Super Challenge - For Loops

Lesson time: 60 Minutes

LESSON OVERVIEW

Students use a mix of different environments to test their knowledge of for loops.

TEACHING SUMMARY

Getting Started

[Introduction](#)

Activity: Super Challenge - For Loops

[Super Challenge - For Loops](#)

LESSON OBJECTIVES

Students will:

- Predict the number of steps needed to increment in each for loop iteration
- Determine how to use a for loop in a way that makes sense for each unique puzzle
- Decompose large complex problems into smaller pieces

GETTING STARTED

Introduction

ACTIVITY

[Super Challenge - For Loops](#)

21

Course 4 | Lesson 21

Super Challenge - Functions with Parameters

Lesson time: 45 Minutes

LESSON OVERVIEW

This challenging stage allows students to hone their skills with functions and parameters to solve complex puzzles with grace and efficiency.

LESSON OBJECTIVES

Students will:

- Look for patterns where they can implement functions
- Utilize parameters to make a single function work for multiple problems

GETTING STARTED

Introduction

This lesson gets complicated rather quickly. If your students had an easy time with previous lessons, then they should be able to keep up just fine. If, however, they found the previous stages difficult, you may want to have them revisit Play Lab: Functions with Paramaters first. In this level, students will not only be using multiple parameters iside their functions, they will also be breaking down large problems into smaller parts to find patterns.

ACTIVITY

Super Challenge - Functions with Parameters

22

Course 4 | Lesson 22

Extreme Challenge - Comprehensive

Lesson time: 45 Minutes

LESSON OVERVIEW

The final stage in this course is intended to test comprehension and transfer of all concepts to blended puzzles.

LESSON OBJECTIVES

Students will:

- Choose from many techniques to find the one that best suits each problem
- Think critically about what they need to accomplish, given the tools that they have

GETTING STARTED

Introduction

If your class has made it this far, their skills are probably already on point. Now, it's time to

ACTIVITY

Extreme Challenge - Comprehensive

Appendix

Code.org K-5 Computer Science

SUPPLEMENTAL DOCUMENTS

Glossary

Code.org CS Fundamentals

Encouraging students to learn and use 'professional' terms enables 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.

TEACHING TIP

Best practice is to introduce the terms with easy-to-understand language, relate the terms to previous experiences, use the terms repeatedly beyond the lesson itself throughout the entire course (and in other situations) when appropriate, and reinforce students' use in oral and written communication.

The following terms are introduced in an unplugged lesson as either a vocabulary word or as a term related to the online Blockly programming interface. Terms are subsequently reinforced in the following online puzzles and activities. Most terms appear multiple times throughout the courses and lessons providing the students with many opportunities to deepen their understanding and assimilate the words into conversations, both in and outside of the classroom.

The course and lesson numbers after each term indicate an unplugged lessons in which the term is emphasized.

VOCABULARY

abstraction

(Course 3: Stage 1 | Course 4: Stage 5)
Pulling out specific differences to make one solution work for multiple problems.

algorithm

(Course 1: Stages 1, 6 | Course 2: Stages 1, 2 | Course 3: Stages 1, 10 | Course 4: Stage 1)
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

(Course 2: Stage 14 | Course 4: Stages 17, 18)
A way of representing information using only two options.

Blockly

(Course 1: Stage 3)

The visual programming language used in Code.org's online learning system for K-5 students.

bug

(Course 1: Stage 5 | Course 2: Stages 9, 10, 11| Course 3: Stage 14)
An error in a program that prevents the program from running as expected.

call (a function)

(Course 3: Stage 4)
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.

code

(Course 1: Stage 2)
One or more commands or algorithm(s) designed to be carried out by a computer. See also: program

command

(Course 1: Stage 2)
An instruction for the computer. Many commands put together make up algorithms and computer programs.

computational thinking

(Course 3: Stage 1)
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

(Course 1: Stage 2)
Using the power of computers to solve problems.

conditionals

(Course 2: Stages 12, 13 | Course 3: Stages 7, 8)
Statements that only run under certain conditions or situations.

crowdsourcing

(Course 3: Stage 19)
Getting help from a large group of people to finish something faster.

data

(Course 2: Stage 14 | Course 3: Stages 1)
Information. Often, quantities, characters, or symbols that are the inputs and outputs of computer programs.

debugging

(Course 1: Stage 5 | Course 2: Stages 9, 10, 11 | Course 3: Stage 14)
Finding and fixing errors in programs.

decompose

(Course 3: Stage 1)
Break a problem down into smaller pieces.

define (a function)

(Course 3: Stage 4)
To add code inside a function so that the program knows what it is supposed to do when the function is called.

digital citizen

(Course 3: Stage 20)
Someone who acts safely, responsibly, and respectfully online.

digital footprint

(Course 2: Stage 18)
The information about someone on the Internet.

DNS (domain name service)

(Course 3: Stage 18)
The service that translates URLs to IP addresses.

DSL/cable

(Course 3: Stage 18)
A method of sending information using telephone or television cables.

event

(Course 1: Stage 15 | Course 2: Stage 15)
An action that causes something to happen.

event-handler

(Course 1: Stage 15 | Course 2: Stage 15)
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.

fiber optic cable

(Course 3: Stage 18)
A connection that uses light to transmit information.

for loop

(Course 4: Stages 8, 9, 10, 11, 20)
A loop with a predetermined beginning, end, and increment (step interval).

function

(Course 3: Stages 4, 5, 6, 9 | Course 4: Stages 12, 13, 14, 15, 16, 21)
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

(Course 3: Stage 4)
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

(Course 3: Stage 4)
The code inside a function that instructs the program on what to do when the function is called.

Internet

(Course 1: Stage 17 | Course 2: Stage 18 | Course 3: Stages 18, 20)
A group of computers and servers that are connected to each other.

IP address

(Course 3: Stage 18)
A number assigned to any item that is connected to the Internet.

iteration

(Course 1: Stage 12 | Course 2: Stage 5)
A repetitive action or command typically created with programming loops.

loop

(Course 1: Stages 12, 13, 14 | Course 2: Stages 5, 6, 7, 8)
The action of doing something over and over again.

packets

(Course 3: Stage 18)
Small chunks of information that have been carefully formed from larger chunks of information.

pattern matching

(Course 3: Stage 1 | Course 4: Stage 5)
Finding similarities between things.

Parameter

(Course 4: Stages 13, 14, 15, 16)
An extra piece of information that you pass to the function to customize it for a specific need.

persistence

(Course 1: Stage 9)
Trying again and again, even when something is very hard.

program

(Course 1: Stage 2 | Course 2: Stage 1, 9 | Course 3: Stage 10)
An algorithm that has been coded into something that can be run by a machine.

programming

(Course 1: Stage 2 | Course 2: Stage 1, 9 | Course 3: Stage 10)
The art of creating a program.

run program

(Course 1: Stage 4)
Cause the computer to execute the commands you've written in your program.

servers

(Course 3: Stage 18)
Computers that exist only to provide things to others.

toolbox

(Course 1: Stage 4)
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.

URL (universal resource locator)

(Course 3: Stage 18)
A relatively easy-to-remember address for calling a web page (like www.code.org).

username

(Course 1: Stage 18)
A name you make up so that you can see or do things on a website, sometimes called a “screen name.”

variable

(Course 3: Stage 4 | Course 4: Stage 4, 5, 6, 7)
A placeholder for a piece of information that can change.

Wi-Fi

(Course 3: Stage 18)
A wireless method of sending information using radio waves.

workspace

(Course 1: Stage 4)
The white area on the right side of Code.org's online learning system where you drag and drop commands to build your program.

Implementation Tips

K-5 Curriculum: Courses 1, 2, 3, & 4

This document offers suggestions for implementing an individual CS Fundamentals course in an elementary school classroom, as well as planning the rollout of all four K-5 courses as a pathway across elementary school grade levels.

SCHEDULING THE LESSONS

Where do you teach the courses? Do you use an existing "special" like 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 set up to run as one lesson per week for a semester (half-year), they can alternatively be run as contiguous lessons for about a month, 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.

UNPLUGGED LESSONS

You can choose to do an entire unplugged lesson, which includes a Getting Started section, Activity, and Wrap-up, and lasts for about 35-45 minutes, or just choose to do the main activity, which usually lasts for 20-25 minutes.

COMPUTERS IN THE CLASSROOM

Here are implementation tips for three common computer use situations in an elementary school classroom:

- Laptop cart
  - Where the grade level teacher leads the class: Increase collaboration by encouraging students to help one another before asking you a question, so that you can attend to the students who are in need of the most help.
- Computer lab
  - Where a media specialist or computer lab teacher leads the class: Split the teaching load by having the computer lab teacher run all of the online puzzle tutorial lessons while you run all of the unplugged lessons in your classroom.
- Computers in the classroom (usually 2 to 6)
  - Use a “centers” approach by mixing an unplugged activity station with a computer-based online puzzle station. In larger classes, you may want to split up components of the unplugged activity into separate stations and use pair programming at the computer station if there aren’t enough computers.

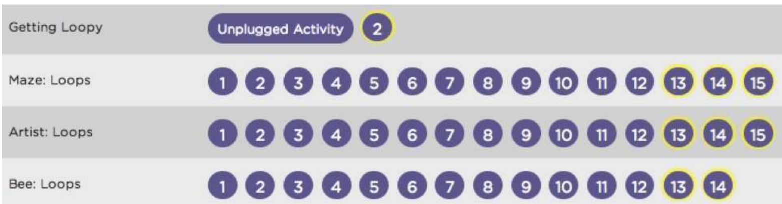
PAIR PROGRAMMING

Show the pair programming video featuring the do’s and don’ts of working collaboratively at a computer.

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 courses to be separated into chunks of lessons that can be taught within a time period, like a week, as a sub-unit.

See this example from *Course 2* in which the unplugged lesson, Getting Loopy, precedes three online puzzle stages.



ROLLOUT

For schools that desire to implement all four courses, the course a student takes depends on their developmental level as well as their experience. While Code.org continues to publish more courses, teachers should strive to introduce material from our third-party partners to create a full pathway. Below is an example of what the first four years of a rollout may look like across grade levels.

	K	1	2	3	4	5
Year 1	Course 1	Course 1	Course 2	Course 2	Course 2	Course 2
Year 2	Course 1	Course 2	Course 2	Course 3	Course 3	Course 3
Year 3	Course 1	Course 2	Course 3	Course 3	Course 4	Course 4
Year 4	Course 1	Course 2	Course 3	Course 4	Course 4	See code.org for 3rd party material

Note: A complete PDF (including course overview and standards) can be found online at: code.org/curriculum/docs/k-5/complete.pdf

The easy, fun, and *active* way to learn computer science!

This book contains lesson plans from CS Fundamentals, Code.org's introduction to computer science for elementary school students and beyond. We have provided lesson plans, worksheets and classroom assessments for each of our courses. We suggest that you follow these lessons in the order presented, but feel free to adapt this plan as needed. You know your classroom best!

- **Course 1** - Intended for early-readers who have little or no previous computer science experience.
- **Course 2**- Intended for readers who have little or no previous computer science experience.
- **Course 3** - Intended for readers who have done Course 2.
- **Course 4** - Intended for readers who have done Course 3.

Teachers are saying:

"I took a year of Pascal in college and learned nothing. I have actually been doing this course with my students, and I have been learning so much."

"I love trying new things! Oh, I should also mention, I am a 57 year old grandmother!"

"After my daughter (who was never particularly interested in math/coding) finished she said, 'Maybe I could be a computer scientist one day.' "

To view the most updated version of these lesson plans and to view the online activities associated with these lesson plans, please visit **http://studio.code.org**.

CODE

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, 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, Reid Hoffman, Salesforce, Sean N. Parker Foundation, Smang Family Foundation, Verizon