Course F

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

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

Teacher Links: Teacher Videos Playlist

Lesson 1: My Robotic Friends

Algorithms | Debugging | Unplugged

Turn your friends into robots and tell them what to do!

Lesson 2: Coding with Comments

Sequencing | Debugging | Loop

In this lesson, you will learn how to write your very own programs!

Lesson 3: Building a Foundation

Unplugged | Persistence | Frustration

Build a structure that can hold a textbook. You might feel frustrated- remember to be persistent!

Lesson 4: Debugging with Scrat

Bug | Debugging | Scrat | Ice Age

Find problems in Ice Age puzzles and practice your debugging skills.

Lesson 5: Creating Art with Code

Artist | Programming | Revise | Private Information

Create beautiful images by programming the Artist.

Lesson 6: My Loopy Robotic Friends

Unplugged | Loop | Repeat

Turn your friends into robots and tell them what to do using loops!

Lesson 7: Drawing Shapes with Loops

Loop | Artist

In this lesson, loops make it easy to make even cooler images with Artist!

Lesson 8: Nested Loops in Maze

Nested Loops | Loops | Bee | Maze

Loops inside loops inside loops. What does this mean? This lesson will teach you what happens when you place a loop inside another loop.

Lesson 9: Nested Loops with Frozen

Loop | Nested Loop | Artist

Anna and Elsa have excellent ice-skating skills, but need your help to create patterns in the ice. Use nested loops to create something super COOL.

Lesson 10: Conditionals with Cards

Conditionals | Unplugged

It's time to play a game where you earn points only under certain conditions!

Lesson 11: Conditionals with the Farmer

Conditionals | Farmer

You will get to tell the computer what to do under certain conditions in this fun and challenging series.

Lesson 12: Functions with Minecraft

Functions | Minecraft

Can you figure out how to use functions for the most efficient code?

Lesson 13: The Power of Words

Common Sense Education | Cyberbullying

Bullying is never okay. This lesson will teach you about what is and isn't okay to say online.

Lesson 14: Envelope Variables

Unplugged | Variable

Envelopes and variables have something in common: both can hold valuable things. Here you will learn what variables are and the awesome things they can do.

Lesson 15: Variables with Artist

Variable | Artist

Don't forget to bring creativity to class! In these puzzles you will be making fantastic drawings using variables.

Lesson 16: Changing Variables with Bee

Variables | Bee

This bee loves variables!

Lesson 17: Changing Variables with Artist

Variable | Artist

In this lesson, you'll make drawings using variables that change as the program runs.

Lesson 18: For Loop Fun

Unplugged | For Loops

You're going to have loads of fun learning about `for` loops!

Lesson 19: For Loops with Bee

For Loop | Bee

Buzz buzz. In these puzzles you will be guiding a bee to nectar and honey using `for` loops!

Lesson 20: For Loops with Artist

For Loop | Artist

Get ready to make your next masterpiece. Here you will be using `for` loops to make some jaw-dropping pictures.

Lesson 21: Learning Sprite Lab

Event | Sprite Lab

Learn how to create and edit sprites.

Lesson 22: Alien Dance Party

Event | Sprite Lab

Practice making games to share with your friends and family.

Lesson 23: Pet Giraffe

Event | Sprite Lab

Make your own pet giraffe!

Lesson 24: Explore Project Ideas

Project | Define | Prepare | Try | Revise | Reflect

Time to get some inspiration! These puzzles will show you a handful of pre-built games and illustrations to help develop your plan for your BIG project.

Lesson 25: The Design Process

Project

Projects this big take time and plenty planning. Here, you will learn about the design process that you'll use to build your own creation.

Lesson 26: Build Your Project

Project

Finally you can start building your project!

Lesson 27: Revise Your Project

Project

Rome wasn't built in a day and your project shouldn't be, either. Take time to edit and revise your project to make it the best it can be.

Lesson 28: Present Your Project

Project

Time to show your work! Here you will be presenting your awesome project to your peers.

Lesson 1: My Robotic Friends

Algorithms | Debugging | Unplugged

Overview

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

Purpose

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

Agenda

Warm Up (5 min)

- **Introduction**

Main Activity (45 min)

- **My Robotic Friends**

Wrap Up (10 min)

- **Journaling**

View on Code Studio

Objectives

Students will be able to:

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

Preparation

□ Watch the **My Robotic Friends - Teacher Video**.

□ Read **My Robotic Friends - Teacher Prep Guide**.

□ Print out one **My Robotic Friends - Symbol Key** per group. This is "code" to be used.

□ **Paper Trapezoid Template - Manipulatives** are provided if your class is not going to use cups.

□ Print out one set of **Stacking Cup Ideas - Manipulatives** per group.

□ Make sure each student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **My Robotic Friends** - Teacher Video
- **My Robotic Friends** - Teacher Prep Guide

For the Students

- **My Robotic Friends Cup Spacing Template** - Template
- **My Robotic Friends** - Unplugged Video (**download**)
- **My Robotic Friends** - Symbol Key
- **Stacking Cup Ideas** - Manipulatives
- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

Warm Up (5 min)

Introduction

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

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

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

Main Activity (45 min)

My Robotic Friends

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

Lesson Tip:

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

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

1. Group the class into pairings or groups of 3. Have the groups pick the first "robot" (there should be enough time for everyone to have their turn). Have the "robots" to leave the classroom until they are called back in.
2. Display **Stacking Cup Ideas - Manipulatives** to the rest of the class. Have each group choose which idea they would like the robot to do. Try to push for an easier idea for the first time, then choose a more complex design later on.
3. Let each group discuss how the stack should be built, then instruct each group to translate the algorithm into the symbols. Make sure each group writes down the symbol algorithm somewhere for the "robot" to read later.
4. Once the groups have all decided on their algorithms, ask the "robots" to come back in. We recommend continuing to display the **My Robotic Friends - Symbol Key** while the robots are building the stack so the student robot remembers what each command means.

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

Lesson Tip:

This activity can be done in one large group, if your class feels uncomfortable with small groups.

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

Wrap Up (10 min)

Journaling

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

Journal Prompts:

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

Standards Alignment

CSTA K-12 Computer Science Standards

▶ **AP** - Algorithms & Programming

Lesson 2: Coding with Comments

Sequencing | Debugging | Loop

Overview

In this set of puzzles, students will begin with an introduction (or review depending on the experience of your class) of Code.org's online workspace. There will be videos pointing out the basic functionality of the workspace including the Run , Reset , and Step buttons. Also discussed in these videos: dragging Blockly blocks, deleting Blockly blocks, and connecting Blockly blocks. Next, students will practice their **sequencing** and **debugging** skills in maze.

Purpose

We recognize that every classroom has a spectrum of understanding for every subject. Some students in your class may be computer wizards, while others haven't had much experience at all. In order to create an equal playing (and learning) field, we have developed these ramp-up lessons. This can be used as either an introduction or a review of how to use Code.org and basic computer science concepts.

Agenda

Warm Up (10 min)

- **Introduction**
- **Vocabulary**

Bridging Activities - Programming (10 min)

- **Unplugged Activity Using Paper Blocks**
- **Preview of Online Puzzles as a Class**

Main Activity (30 min)

- **Course F Online Puzzles - Website**

Wrap Up (10 min)

- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Order movement commands as sequential steps in a program.
- Modify an existing program to solve errors.
- Break down a long sequence of instructions into the largest repeatable sequence.

Preparation

☐ Play through the **Course F Online Puzzles - Website** to find any potential problem areas for your class.

☐ Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Loop** - The action of doing something over and over again.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

- **Programming** - The art of creating a program.

Teaching Guide

Warm Up (10 min)

Introduction

Students will either be learning a lot of new concepts or reviewing a lot of basic concepts. Based on your class's experience, you can cover the following vocabulary or move on to a bridging activity. We recommend using the following words in sentences if the definitions aren't explicitly covered.

Vocabulary

This lesson has four new and important vocabulary words:

- **Program** - Say it with me: Pro - Gram An algorithm that has been coded into something that can be run by a machine.
- **Programming** - Say it with me: Pro - Gramm - ing The art of creating a program.
- **Bug** - Say it with me: Bug An error in a program that prevents the program from running as expected.
- **Debugging** - Say it with me: De - Bugg - ing Finding and fixing errors in programs.
- **Loop** - Say it with me: Loo-p The action of doing something over and over again

Bridging Activities - Programming (10 min)

This activity will help bring the unplugged concepts from "My Robotic Friends" into the online world that the students are moving into. Choose **one** of the following to do with your class:

Unplugged Activity Using Paper Blocks

Similar to "My Robotic Friends", have the students in your class pair up. Pass out multiple fill 1 and move ___ blocks from the **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives** to each pair. Have each pair of students draw a design on a four by four graph from **Graph Paper Programming - Worksheet** . Next, have the students work together to write the program needed to draw this design using the paper Blockly blocks. The students will need to write up , down , right , or left on the move ___ block. Make sure the students know that the program goes from top to bottom and the blocks need to touch!

Preview of Online Puzzles as a Class

Pull up a puzzle from **Course F Online Puzzles - Website** . We recommend puzzle 6 for this activity. Break up the students into groups of three or four. Have them "program" Red, the angry bird, to get to the pig using arrows from "My Robotic Friends".

↑ - Pick Up Cup

↓ - Put Down Cup

→ - Move 1/2 Cup Width Forward

← - Move 1/2 Cup Width Backward

↻ - Turn Cup Right 90°

↺ - Turn Cup Left 90°

Once all the groups have an answer, discuss the path as a class.

Main Activity (30 min)

Course F Online Puzzles - Website

Teachers play a vital role in computer science education and supporting a collaborative and vibrant classroom environment. During online activities, the role of the teacher is primarily one of encouragement and support. Online lessons are meant to be student-centered, so teachers should avoid stepping in when students get stuck. Some ideas on how to do this are:

- Utilize **Pair Programming - Student Video** whenever possible during the activity.
- Encourage students with questions/challenges to start by asking their partner.
- Unanswered questions can be escalated to a nearby group, who might already know the solution.
- Remind students to use the debugging process before you approach.
- Have students describe the problem that they're seeing. What is it supposed to do? What does it do? What does that tell you?
- Remind frustrated students that frustration is a step on the path to learning, and that persistence will pay off.
- If a student is still stuck after all of this, ask leading questions to get the student to spot an error on their own.

Teacher Tip:

Show the students the right way to help classmates:

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

Wrap Up (10 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel about today's lesson?
- List some of the bugs you found in your programs today.
- What was your favorite puzzle to complete? Draw your favorite character completing a puzzle.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 3: Building a Foundation

Unplugged | Persistence | Frustration

Overview

New and unsolved problems are often pretty hard. If we want to have any chance of making something creative, useful, and clever, then we need to be willing to attack hard problems even if it means failing a few times before we succeed. In this lesson, students will be building a structure with common materials. The structure will be tested on its ability to hold a textbook for more than ten seconds. Most students will not get this right the first time, but it's important they push through and keep trying.

Purpose

This lesson teaches that failure is not the end of a journey, but a hint for how to succeed. The majority of students will feel frustrated at some point in this lesson, but it's important to emphasize that failure and frustration are common steps to creativity and success.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Try, Try Again**

Main Activity (20 min)

- **Building a Foundation**

Wrap Up (10 min)

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

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Outline steps to complete a structural engineering challenge.
- Predict and discuss potential issues in structure creation.
- Build a structure based on team plan.
- Revise both the plan and the structure until they satisfy the challenge.

Preparation

□ Watch the **Building a Foundation - Teacher Video**.

□ Watch the **Building a Foundation - Lesson in Action Video**.

□ Print **Building a Foundation - Teacher Prep Guide**.

□ Gather enough building elements (marshmallows or gumdrops with toothpicks or popsicle sticks) for each group. You don't have to give any certain amount; just make sure you put some limit on materials.

□ Give a **Think Spot Journal - Reflection Journal** to each student.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Building a Foundation** - Unplugged Video (**download**)
- **Building a Foundation** - Teacher Video
- **Building a Foundation** - Lesson in Action Video
- **Building a Foundation** - Teacher Prep Guide

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Frustrated** - Feeling annoyed or angry because something is not the way you want it.
- **Persistence** - Trying again and again, even when something is very hard.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

Persistence - Say it with me: Per-sis-tence

Trying again and again, even when something is very hard

Try, Try Again

- Does everyone get everything right the first time?
- When I was a baby learning to walk, did I stand up and run off on my first try?
- Sometimes, the best and most useful things to do are the hardest to learn.
  - It can take a while to learn hard things
  - If you don't do something well at first, does it mean that you never will?
  - Can you think of something that was hard at first, but that you can now do pretty easily?
    - Walking
    - Talking
    - Riding a bike
- When you fail at doing something, you get a hint at what went wrong. You just need to look for it.
  - If your bike tips over, next time you need to work on balance.
  - If you're filling a balloon and it pops, next time you need less air.
- Think of the mistakes as chances to learn how to do something better next time.

Lesson Tip

Here are some great resources to prep your class with the concept of persistence before you turn them loose on this project:

- **Mouse Wants a Cracker**
- **Fall 7 Times, Stand Up 8**
- **Never Ever Give Up**
- **If You Quit Too Soon**

Main Activity (20 min)

Building a Foundation

Have you ever started on a task, then discovered that it was much harder than you thought it would be? Hard tasks can make us want to give up, but if we stick to our goal and keep trying, then we just might make something better than we've ever made before!

In this challenge, we'll work to construct towers that are strong enough to hold a textbook for at least 10 seconds, using everyday materials.

Rules:

- Use only the supplies provided to build a tower.
- The tower can be any shape, but it has to be at least as tall as the paper cup.
- The tower must support the weight of a book for a full 10 seconds.

Directions:

1. Divide students into groups of three or four.
2. Explain the rules of the challenge, given above.
3. Provide each group with limited supplies and make it known that they will get no more.
4. Challenge the class to think ahead to the problem and plan out their method of building their first tower.

5. Encourage students to begin building, then have them alert you when they think they've met the challenge described by the rules.
6. Test each structure. Is it taller than the cup? Does it hold a book?
7. If not, have students enter a cycle of planning, fixing, testing, and planning again until the challenge has been met.
8. Congratulate the students as they succeed and take pictures of the successful towers!

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 (10 min)

Flash Chat: What did we learn?

- Were you proud of what you made?
- Do you think you could make a tower as tall as a chair that could hold a person?
  - How many gumdrops do you think you would need?
- Was there a time that you thought about giving up?
  - How did you get past that feeling?

Lesson Tip

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

Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future. We provide a **Think Spot Journal - Reflection Journal** as a basic template for students to use as their daily journal.

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw a picture of your structure.
- What were some problems you ran into while building? How did you fix these problems?

Extended Learning

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

Try It Again!

Try doing the same activity with different materials.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 4: Debugging with Scrat

Bug | Debugging | Scrat | Ice Age

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 blocks that are out of order.

Purpose

Students in your class might become frustrated with this lesson because of the essence of debugging. **Debugging** is a concept that is very important to computer programming. Computer scientists have to get really good at facing the bugs in their own programs. Debugging forces the students to recognize problems and overcome them while building critical thinking and problem solving skills.

Agenda

Warm Up (15 min)

- **Introduction**
- **Vocabulary**

Main Activity (30 min)

- **Course F Online Puzzles - Website**

Wrap Up (5 - 10 min)

- **Journaling**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Predict where a program will fail.
- Modify an existing program to solve errors.
- Reflect on the debugging process in an age-appropriate way.

Preparation

□Play through the **Course F Online Puzzles - Website** to find any potential problem areas for your class.

□(Optional) Pick a couple of puzzles to do as a group with your class.

□Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

□Make sure every student has a**Think Spot Journal - Reflection Journal**.

□Review **Debugging Recipe - Student Handout** with the class.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal
- **Debugging Recipe** - Student Handout

Vocabulary

- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.

Teaching Guide

Warm Up (15 min)

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.

Vocabulary

This lesson has three new and important vocabulary words:

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

Say:

Debugging is a process. First, you must recognize that there is an error in your program. You then work through the program step by step to find the error. Try the first step, did it work? Then the second, how about now? If you make sure that everything is working line by line, then when you get to the place that your code isn't doing what it's supposed to, you know that you've found a bug. Once you've discovered your bug, you can work to fix (or "debug") it!

If you think it will build excitement in the class you can introduce the character of today's puzzles, Scrat from Ice Age. If students aren't familiar with Scrat, **show some videos** of the quirky squirrel running into trouble.

Main Activity (30 min)

Course F Online Puzzles - Website

Before letting the students start on the computer, remind them of the advantages of**Pair Programming - Student Video** and asking their peers for help. Sit students in pairs and recommend they ask at least two peers for help before they come to a teacher.

As mentioned in the purpose of this lesson, make sure the students are aware that they will face frustrating puzzles. Tell them it is okay to feel frustrated, but it is important to work through the problem and ask for help. As the students work through the puzzles, walk around to make sure no student is feeling so stuck that they aren't willing to continue anymore.

Wrap Up (5 - 10 min)

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- What kind of bugs did you find today?

- Draw a bug you encountered in one of the puzzles today. What did you do to "debug" the program?

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.

When other students are debugging, make sure that the criticisms are constructive. If this could be a problem for your class, go over respectful debugging before this activity by role playing with another student.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▸ **AP** - Algorithms & Programming

Lesson 5: Creating Art with Code

Artist | Programming | Revise | Private Information

Overview

In this lesson, students will take control of the Artist to complete drawings on the screen. This Artist stage will allow students to create images of increasing complexity using new blocks like move forward by 100 pixels and turn right by 90 degrees .

Purpose

Building off of the students' previous experience with sequencing, this lesson will work to inspire more creativity with coding. The purpose of this lesson is to solidify knowledge on sequencing by introducing new blocks and goals. In this case, students learn more about pixels and angles using the new blocks, while still practicing their sequencing skills. Also, students will be able to visualize new goals such as coding the Artist to draw a square.

Agenda

Warm Up (10 min)

  Introduction

Main Activity (30 min)

  Course F Online Puzzles - Website

Wrap Up (10 - 15 min)

  Journaling

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Create a program to complete an image using sequential steps.
- Break complex shapes into simple parts.

Preparation

□Play through the **Course F Online Puzzles - Website** in stage 6 to find any potential problem areas for your class.
□Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
□(Optional) Obtain protractors for your class to visualize the angles they must use to complete the puzzles.
□Print one **Turns & Angles - Student Handout** for each student.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal
- **Artist Introduction** - Student Video
- **Turns & Angles** - Student Video
- **Turns & Angles** - Student Handout

Teaching Guide

Warm Up (10 min)

Introduction

Show the students one or both of the following videos as an introduction to angles:

Artist Introduction - Student Video (1.5 minutes long)

Turns & Angles - Student Video (2 minutes long)

Use the **Turns & Angles - Student Handout** to show the students interior versus exterior angles for different shapes. This document can be used as a hand out or you can choose to print it out as a poster for students to refer to.

Ask:

Discuss the square and triangle shapes from the document.

- How would you code a computer to draw that shape?
- What order do the instructions need to be in?

Tell the students that in these puzzles they will be moving a character who leaves a line everywhere he goes. The students will be writing code that gets the character to draw various shapes, including a square.

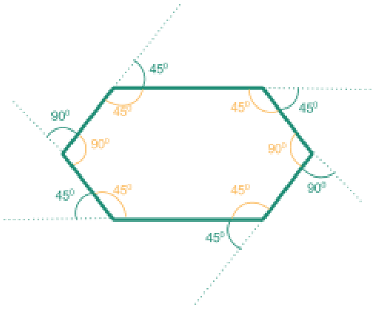
Main Activity (30 min)

Course F Online Puzzles - Website

In this set of puzzles, the artist will no longer be constrained to 90 degree angles. Having physical protractors available can be help students better visualize the angles they need. Otherwise, the stage provides images of the angles as the student selects which angle to use. (Please note: Angle choices are limited to two inside of the dropdown menu, reducing the number of options students have to work through.)

Before sending the students to the computers to work on the puzzles, it might be beneficial to give a brief presentation of how to use the tools in this level. We recommend puzzle 5 as a good puzzle to show how to use the protractor online.

The eighth puzzle asks the students to draw a 6 sided polygon. This might be challenging for some students. We recommend getting the students to try a few times, ask a peer, then ask the teacher for help. Below is an image that might be helpful for the students.



Wrap Up (10 - 15 min)

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- What are the interior angles that make up a square. What about for a triangle?
- Sketch a simple shape on your paper and imagine the code used to draw it. Can you write that code out next to the shape?

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.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 6: My Loopy Robotic Friends

Unplugged | Loop | Repeat

Overview

Building on the initial "My Robotic Friends" activity, students learn to use loops when programming their robots in order to build bigger structures more efficiently.

Purpose

This lesson serves as a reintroduction to loops, using the now familiar set of "robot" programming instructions. Students will develop critical thinking skills by looking for patterns of repetition in the movements of classmates and determining how to simplify those repeated patterns using loops.

Agenda

- **Warm Up (10 min)**
  - **My Robotic Friends Review**
- **Activity (30 min)**
  - **Introduction and Modeling**
  - **Looping Your Robots**
- **Wrap Up (5 min)**
  - **Journaling**
- **Extension Activities**

View on Code Studio

Objectives

Students will be able to:

- Identify repeated patterns in code that could be replaced with a loop
- Write instructions that use loops to repeat patterns

Preparation

☐Watch the **My Loopy Robotic Friends - Teacher Video**.

☐(Optional) Print out one **My Robotic Friends - Symbol Key** per group or 4 students. Alternatively, find a place to display this information where students can reference throughout the lesson.

☐Prepare a stack of 20 paper cups for each group of 4 students. OR

☐(Optional) print and cut out **Paper Trapezoid Template - Manipulatives** for each group if your class is not going to use cups.

☐Print out one set of **Stacking Cup Ideas - Manipulatives** per group.

☐Make sure each student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **My Loopy Robotic Friends** - Teacher Video

For the Students

- **My Robotic Friends Cup Spacing Template** - Template
- **Paper Trapezoid Template** - Manipulatives

- **Stacking Cup Ideas** - Manipulatives
- **My Robotic Friends** - Symbol Key

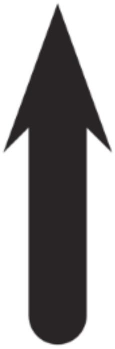
Vocabulary

- **Loop** - The action of doing something over and over again.
- **Program** - An algorithm that has been coded into something that can be run by a machine.
- **Programming** - The art of creating a program.
- **Repeat** - Do something again

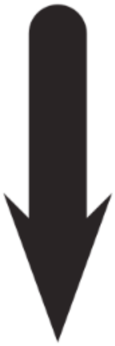
Teaching Guide

Warm Up (10 min)

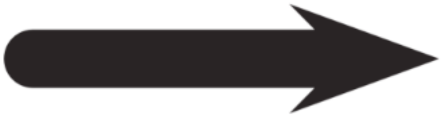
My Robotic Friends Review



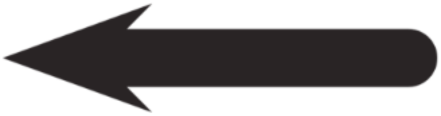
Pick Up Cup



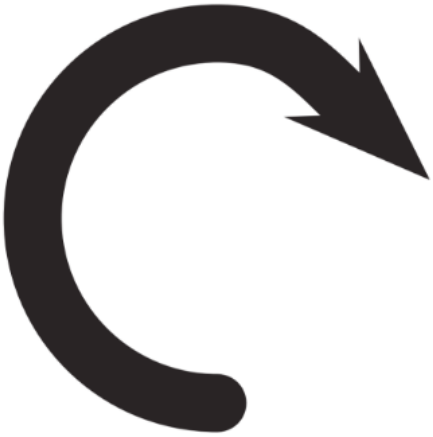
Put Down Cup



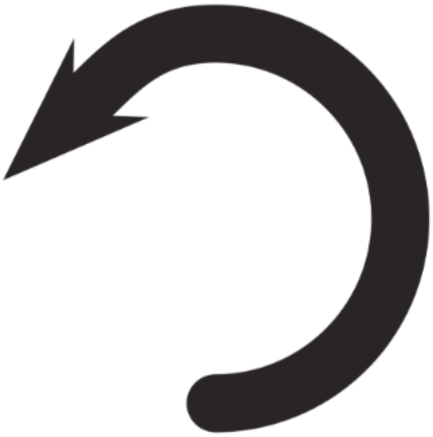
Step Forward



Step Backward



Turn Cup Right 90°



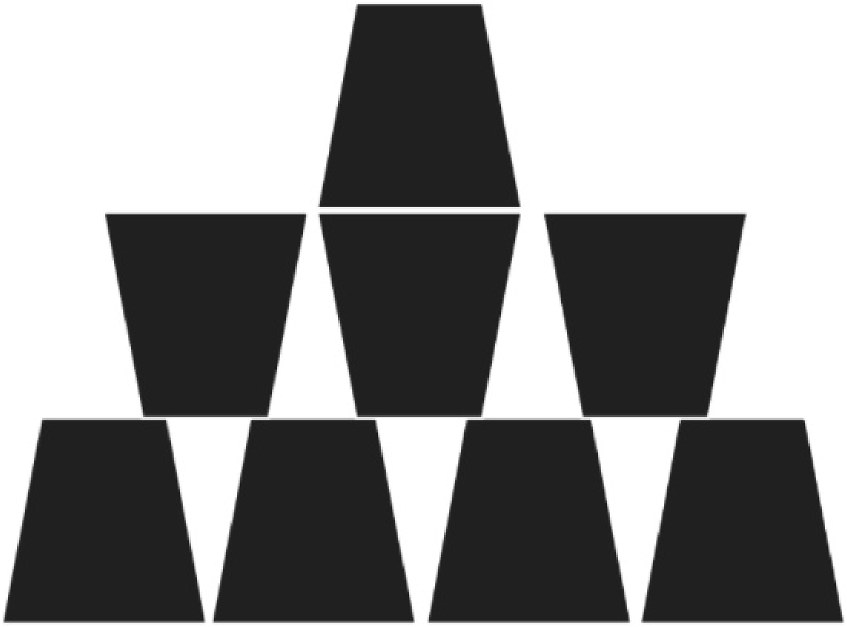
Turn Cup Left 90°

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

Display: Show the **My Robotic Friends - Symbol Key** that we used in My Robotic Friends. For each of the six symbols, ask students to show you what it looks like for a robot to follow that instruction.

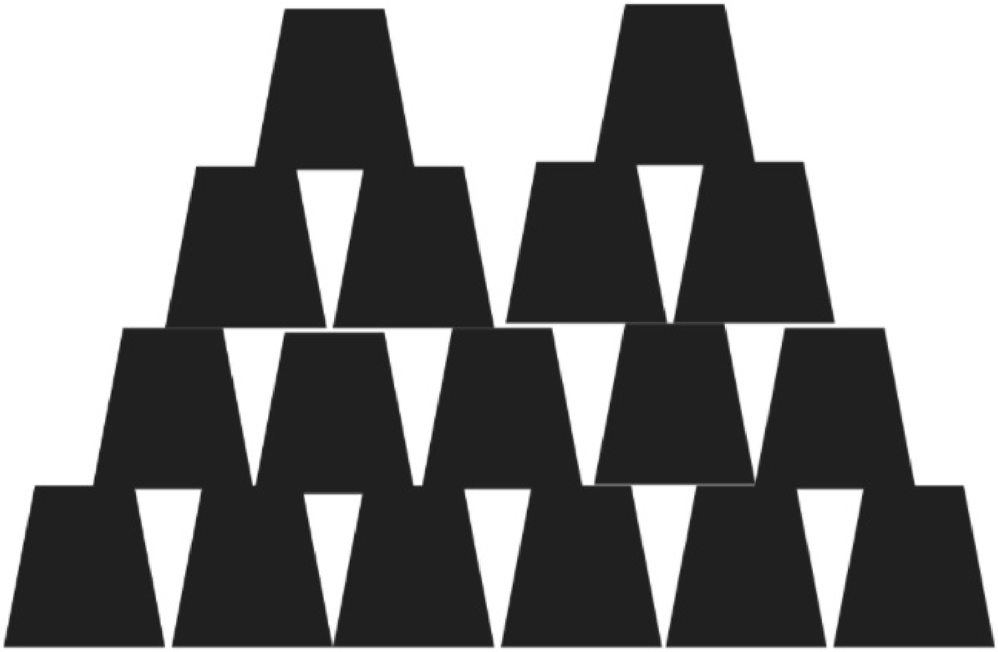
Model: With the class together as a group, pull an easy puzzle from the "My Robotic Friends" Cup Stack Pack and program with each other as a reminder of rules and terminology.

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



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

Display: Now, bring up this image:



What is the reaction of the class?

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

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

Activity (30 min)

Introduction and Modeling

Set Up: Have stacks of cups or cut paper trapezoids available for groups.

Display: Take the program from one of your previous cup stacks and display it for the class, or use the one below.

↑ → → ↓ ← ←

↑ → → → → ↓ ← ← ← ←

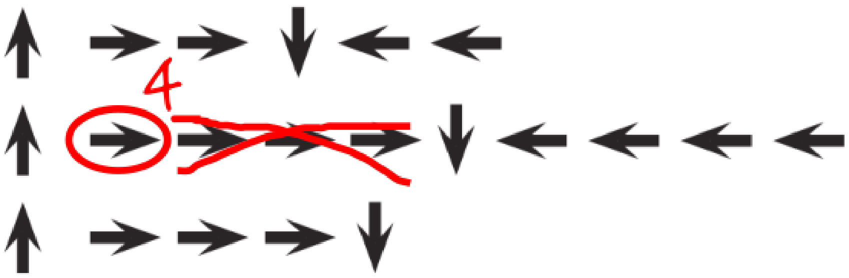
↑ → → → ↓

Think: Ask students to think quietly about where in this program they can find a pattern of instructions that repeat uninterrupted (one repetition after another).

Pair: Turn to a neighbor and share one of the repeating patterns you found.

Share: Ask a few students to share out the patterns they identified. Try to pull out different approaches to grouping patterns. For each pattern, ask students to identify how many times the pattern repeats.

Model: Using one of the repeating patterns that the class identified, model how Circle the instruction or pattern that repeats, write the number of loops near that circle, then cross out the rest of the arrows.



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

Looping Your Robots

Group: Place students into groups of 4. Each group should then further break down into two pairs - each pair will develop their own program "run" on the other pair.

Distribute: Give each group one stack of cups or paper cutouts.

Display: Show **Stacking Cup Ideas - Manipulatives** to the class or hand out individual copies for groups to use.. Have each pair (not group) choose which idea they would like their robot to do. Encourage pairs to select a more complicated pattern this time around.

Discuss: Let each group discuss how the stack should be built, then instruct each group to translate the algorithm into the symbols. Make sure each group writes down the symbol algorithm somewhere for the "robot" to read later. As students are working on their programs, remind them to be on the lookout for opportunities to replace a repeating pattern with a loop.

Do: When groups have finished their instructions, have each pair take turns "running" their code with another pair. Remind students to be on the lookout for bugs in their code, but not to interrupt a robot until it's finished running the program.

Discuss: When all of the pairs have had a chance to run their programs, ask a few to share their solutions with the class. Use this opportunity to discuss how groups came up with different solutions to the same puzzle. In particular, you might ask of each program:

- How did they identify the loops?
- Are there other ways those loops could have been written?

Teaching Tip

Looking for Loops: Be sure to keep your eyes open for students using loops. Try to avoid correcting their overall algorithms or prescribing a solution, but feel free to direct students towards patterns that could be shortened by using a repeat circle.

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

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

- How much shorter is the program with loops than it would be without?
- Is the program easier to understand with loops, or written out longhand? Why?

Wrap Up (5 min)

Journaling

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

Flash Chat:

Here are some possible topics:

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

Journal Prompts:

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

Extension Activities

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

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 7: Drawing Shapes with Loops

Loop | Artist

Overview

Watch student faces light up as they make their own gorgeous designs using a small number of blocks and digital stickers! This lesson builds on the understanding of loops from previous lessons and gives students a chance to be truly creative. This activity is fantastic for producing artifacts for portfolios or parent/teacher conferences.

Purpose

This series highlights the power of loops with creative and personal designs.

Offered as a project-backed sequence, this progression will allow students to build on top of their own work and create amazing artifacts.

Agenda

Warm Up (15 min)
- **Introduction**

Main Activity (30 min)
- **Course F Online Puzzles - Website**

Wrap Up (15 min)
- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Identify the benefits of using a loop structure instead of manual repetition.
- Differentiate between commands that need to be repeated in loops and commands that should be used on their own.

Preparation

□Play through the **Course F Online Puzzles - Website** corresponding to this course to find any potential problem areas for your class.
□Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
□Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Introduction

Students should have had plenty of introduction to loops at this point. Based on what you think your class could benefit from, we recommend:

- Creating a new stack design with loops just like in "My Loopy Robotic Friends"
- Reviewing how to use Artist by playing through a puzzle from "Programming in Artist"
- Previewing a puzzle from this lesson

All of these options will either review loops or the artist, which will help prepare your class for fun with the online puzzles!

Main Activity (30 min)

Course F Online Puzzles - Website

Some students may discover where to add `repeat` loops by writing out the program without loops then circling sections of repetitions. If the students in your class seem like they could benefit from this, have them keep paper and pencils beside them at their machines. Students might also enjoy drawing some of the shapes and figures on paper before they program it online. (When drawing stamps, it can be easier to symbolize those with simple shapes like circles and squares.)

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- What was the coolest shape or figure you programmed today? Draw it out!
- What is another shape or figure you would like to program? Can you come up with the code to create it?

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **AP** - Algorithms & Programming

Lesson 8: Nested Loops in Maze

Nested Loops | Loops | Bee | Maze

Overview

In this online activity, students will have the opportunity to push their understanding of loops to a whole new level. Playing with the Bee and Plants vs Zombies, students will learn how to program a loop to be inside of another loop. They will also be encouraged to figure out how little changes in either loop will affect their program when they click Run .

Purpose

In this introduction to **nested loops**, students will go outside of their comfort zone to create more efficient solutions to puzzles.

In earlier puzzles, loops pushed students to recognize repetition. Here, students will learn to recognize patterns **within** repeated patterns to develop these **nested loops**. This stage starts off by encouraging students try to solve a puzzle where the code is irritating and complex to write out the long way. After a video introduces **nested loops**, students are shown an example and asked to predict what will happen when a loop is put inside of another loop. This progression leads into plenty of practice for students to solidify and build on their understanding of looping in programming.

Agenda

- **Warm Up (10 min)**
  - **Introduction**
- **Main Activity (30 min)**
  - **Course F Online Puzzles - Website**
- **Wrap Up (15 min)**
  - **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Break complex tasks into smaller repeatable sections.
- Recognize large repeated patterns as made from smaller repeated patterns.
- Identify the benefits of using a loop structure instead of manual repetition.

Preparation

□ Play through **Course F Online Puzzles - Website** to find any potential problem areas for your class.

□ Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

□ Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

Briefly review with the class what loops are and why we use them.

- What do loops do?
  - Loops repeat a set of commands. (see vocabulary on command if students don't recognize it)
- How do we use loops?
  - We use loops to create a pattern made of repeated actions.

Tell the class that they will now be doing something super cool: using loops inside loops. Ask the class to predict what kinds of things we would be using a loop inside of a loop for.

"If a loop repeats a pattern, then looping a loop would repeat a pattern of patterns!"

Students don't need to understand this right away, so feel free to move on to the online puzzles even if students still seem a little confused.

Main Activity (30 min)

Course F Online Puzzles - Website

We highly recommend **Pair Programming - Student Video** in this lesson. This may not be an easy topic for the majority of your students. Working with a partner and discussing potential solutions to the puzzles might ease the students' minds.

Also, have paper and pencils nearby for students to write out their plan before coding. Some puzzles have a limit on the number of certain blocks you can use, so if students like to write out the long answer to find the repeats, paper can be useful.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel about today's lesson?
- What is a nested loop?
- Can you draw a puzzle that would use a nested loop? Try coding the solution to your own puzzle.

Standards Alignment

CSTA K-12 Computer Science Standards

▶ **AP** - Algorithms & Programming

Lesson 9: Nested Loops with Frozen

Loop | Nested Loop | Artist

Overview

Now that students know how to layer their loops, they can create so many beautiful things. This lesson will take students through a series of exercises to help them create their own portfolio-ready images using Anna and Elsa's excellent ice-skating skills!

Purpose

In this series, students will get practice nesting loops while creating images that they will be excited to share.

Beginning with a handful of instructions, students will make their own decisions when it comes to creating designs for repetition. They will then spin those around a variety of ways to end up with a work of art that is truly unique.

Agenda

Warm Up (15 min)

Introduction

Main Activity (30 min)

Course F Online Puzzles - Website

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Describe when a loop, nested loop, or no loop is needed.
- Recognize the difference between using a loop and a nested loop.
- Break apart code into the largest repeatable sequences using both loops and nested loops.

Preparation

□ Play through the **Course F Online Puzzles - Website** corresponding to this lesson to find and potential problem areas for your class.
□ Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
□ Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Introduction

Ask the class to discuss the last set of puzzles.

- What did they like/dislike?
- Which puzzles were hard? Why?
- Which puzzles were easy? Why?
- If you were to teach nested loops to a friend, what would you say to help them understand?

If there's time, give an introduction to the main characters of today's puzzles, Anna and Elsa from Frozen. Give the class the sister's back story if the class doesn't already know. To build excitement, tell the class they will be using nested loops to make some fantastic drawings with Anna and Elsa's ice skates!

Main Activity (30 min)

Course F Online Puzzles - Website

This set of puzzles is set up as a progression. This means every puzzle builds a foundation for the next puzzle. Students will enjoy making more and more interesting designs by making small and simple changes to code they have already written.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- When do you use a loop? When do you use a nested loop?
- Thought exercise: Can you make everything a nested loop can with just a normal loop? Can you draw out an example?
  - Answer: Yes, you can, but it is a lot more difficult. Nested loops make programs simpler.

Standards Alignment

CSTA K-12 Computer Science Standards

▶ **AP** - Algorithms & Programming

Lesson 10: Conditionals with Cards

Conditionals | Unplugged

Overview

This lesson demonstrates how conditionals can be used to tailor a program to specific information. We don’t always have all of the information we need when writing a program. Sometimes you will want to do something different in one situation than in another, even if you don't know what situation will be true when your code runs. That is where conditionals come in. Conditionals allow a computer to make a decision, based on the information that is true any time your code is run.

Purpose

One of the best parts of teaching **conditionals** is that students already understand the concept from their everyday lives.

This lesson merges computer science into the real world by building off of their ability to tell if a condition is true or false. Students will learn to use `if` statements to declare when a certain command should be run, as well as `if / else` statements to declare when a command should be run and what do run otherwise. Students may not recognize the word **conditionals**, but most students will understand the idea of using "if" to make sure that some action only occurs when it is supposed to.

Agenda

- **Warm Up (20 min)**
  - **Vocabulary**
  - **Introduction**
- **Main Activity (20 min)**
  - **Conditionals with Cards Sample Program - Teacher Prep Guide**
- **Wrap Up (15 min)**
  - **Flash Chat: What did we learn?**
  - **Journaling**
- **Assessment (5 min)**
  - **Conditionals with Cards - Assessment**
- **Extended Learning**

View on Code Studio

Objectives

Students will be able to:

- Define circumstances when certain parts of a program should run and when they shouldn't.
- Determine whether a conditional is met based on criteria.
- Traverse a program and predict the outcome, given a set of input.

Preparation

□Watch the **Conditionals with Cards - Teacher Video**.

□Watch the **Conditionals with Cards - Lesson in Action Video**.

□Gather decks of cards or something similar.

□One **Conditionals with Cards Sample Program - Teacher Prep Guide** for the class to look at.

□Print one **Conditionals with Cards - Assessment** for each student.

□Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Conditionals with Cards** - Unplugged Video (**download**)
- **Conditionals with Cards** - Teacher Video
- **Conditionals with Cards** - Lesson in Action Video
- **Conditionals with Cards Sample Program** - Teacher Prep Guide
- **Conditionals with Cards** - Assessment
- **Conditionals with Cards** - Assessment Video

- **Conditionals with Cards** - Assessment Answer Key

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Conditionals** - Statements that only run under certain conditions.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

Conditionals - Say it with me: Con-di-shun-uls

Statements that only run under certain conditions.

Introduction

- We can start this lesson off right away
  - Let the class know that if they can be completely quiet for thirty seconds, you will do something like:
    - Sing an opera song
    - Give five more minutes of recess
    - Do a handstand
  - Start counting right away.
  - If the students succeed, point out that they succeeded, so they get the reward.
  - Otherwise, point out that they were not completely quiet for a full thirty seconds, so they do not get the reward.
- Ask the class "What was the condition of the reward?"
  - The condition was IF you were quiet for 30 seconds
    - If you were, the condition would be true, and you would get the reward.
    - If you weren't, the condition would be false, so the reward woud not apply.
  - Can we come up with another conditional?
    - If you can guess my age correctly, the class can give you applause.
    - If I know an answer, I can raise my hand.
    - What examples can you come up with?
- Sometimes, we want to have an extra condition, in case the "IF" statement is not true.
  - This extra condition is called an "ELSE" statement
  - When the "IF" condition isn't met, we can look at the "ELSE" for what to do
    - Example: IF I draw a king from this deck of cards, everybody claps. Or ELSE, everyone says "Awwwwwwe."
    - Let's try it. (Draw a card and see if your class reacts appropriately.)
  - Ask the class to analyze what just happened.
    - What was the IF?
    - What was the ELSE?
    - Which condition was met?
  - Believe it or not, we have even one more option.
    - What if I wanted you to clap if I draw a 7, or else if I draw something less than seven you say "YAY," or else you say "Awwwwwwwe"?
      - This is why we have the terms If, Else-If, and Else.
      - If is the first condition
      - Else-If gets looked at only if the "If" isn't true.
      - Else gets looked at only if nothing before it is true.

Now let's play a game.

Main Activity (20 min)

Conditionals with Cards Sample Program - Teacher Prep Guide

Directions:

- Create a few programs with your class that depend on things like a card's suit, color, or value to award or subtract points. You can write the program as an algorithm, pseudocode, or actual code.

Here is a sample algorithm:

```
if (CARD is RED)
    Award YOUR team 1 point

Else
    Award OTHER team 1 point
```

Here is a sample of the same program in pseudocode:

```
If (card.color == RED){
    points.yours = points.yours + 1;
}

Else {
    points.other = points.other + 1;
}
```

- Decide how you want to split your class into teams.
- Each team should have a pile of cards (at least as many cards as team members) nearby.
- Put one of your “Programs” up on the board for all to see.
- Have the teams take turns drawing cards and following the program to see how many points they score in each round.
- Play several times with several different programs to help the students really understand conditionals.

Once the class has had some practice, you can encourage students to nest conditionals inside one another. Make sure they understand that if the card is red, YOUR team is awarded 1 point, and then **nothing else happens**, since the condition was met:

```
If (CARD is RED){
    Award YOUR team 1 point

Else
    If (CARD is higher than 9)
        Award OTHER team 1 point
    Else
        Award YOUR team the same number of points on the card
```

Here is the same program in pseudocode:

```
If (card.color == RED ){
    points.yours = points.yours + 1;
}
Else {
    if (card.value > 9){
        points.other = points.other + 1;
    }
    Else {
        points.yours = points.yours + card.value;
    }
}
```

Wrap Up (15 min)

Flash Chat: What did we learn?

- If you were going to code this up in Blocky, what would you need to add around your conditionals to let the code run more than one time? (A loop)

- What other things do you do during the day under certain conditions?
- If you are supposed to do something when the value of a card is more than 5, and you draw a 5, do you meet that condition?
- Notice that conditions are either "True" or "False." There is no assessment of a condition that evaluates to "Banana."
- When you need to meet several combinations of conditions, we can use something called "nested conditionals."
  - What do you think that means?
  - Can you give an example of where we saw that during the game?
- What part of that game did you like the best?

Lesson Tip

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

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a conditional? How did you use a conditional today?
- What are some of the conditionals you used today? Can you come up with some more that you would use with a deck of cards?

Assessment (5 min)

Conditionals with Cards - Assessment

Hand out the assessment worksheet and allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities. Here's a **Conditionals with Cards - Assessment Video** to watch as a guide.

Extended Learning

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

True/False Tag

- Line students up as if to play**Red Light / Green Light** .
- Select one person to stand in front as the Caller.
- The Caller chooses a condition and asks everyone who meets that condition to take a step forward.
  - If you have a red belt, step forward.
  - If you are wearing sandals, take a step forward.
- Try switching it up by saying things like "If you are**not** blonde, step forward."

Nesting

- Break students up into pairs or small groups.
- Have them write if statements for playing cards on strips of paper, such as:
  - the suit is clubs
  - the color is red
- Have students create similar strips for outcomes.
  - Add one point
  - Subtract one point

- Once that's done, have students choose three of each type of strip and three playing cards, paying attention to the order selected.
- Using three pieces of paper, have students write three different programs using only the sets of strips that they selected, in any order.
    - Encourage students to put some if statements inside other if statements.
- Now, students should run through all three programs using the cards that they drew, in the same order for each program.
    - Did any two programs return the same answer?
    - Did any return something different?

Standards Alignment

CSTA K-12 Computer Science Standards

▶ **AP** - Algorithms & Programming

Lesson 11: Conditionals with the Farmer

Conditionals | Farmer

Overview

This lesson introduces students to while loops, until loops, and if / else statements. **While loops** are loops that continue to repeat commands as long as a condition is true. While loops are used when the programmer doesn't know the exact number of times the commands need to be repeated, but the programmer does know what condition needs to be true in order for the loop to continue looping. Until loops keep going until something specific is true. If / Else statements offer flexibility in programming by running entire sections of code only if something is true, otherwise it runs something else.

Purpose

A basic understanding of conditionals is a recommended prerequisite for Course E. We created this introduction to give a review for the students already familiar to conditionals and allow practice for the students that are just learning. If you find that the understanding of conditionals varies widely in your classroom, we recommend a strategic pairing of students when completing this online lesson.

Agenda

Warm Up (15 min)

- **Introduction**

Main Activity (30 min)

- **Course F Online Puzzles - Website**

Wrap Up (15 min)

- **Journaling**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Define circumstances when certain parts of a program should run and when they shouldn't.
- Determine whether a conditional is met based on criteria.

Preparation

□ Play through puzzles in **Course F Online Puzzles - Website** to find any potential problem areas for your class.

□ Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

□ Make sure every student has a **Think Spot Journal - Reflection Journal**

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Condition** - A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored.
- **Conditionals** - Statements that only run under certain conditions.
- **While Loop** - A loop that continues to repeat while a condition is true.

Teaching Guide

Warm Up (15 min)

Introduction

Gather the class together and ask two volunteers to walk straight in some direction in the classroom. If they encounter a chair out of place, they must step over it. If they reach a wall, they must sit down.

Once all of the students are sitting down, ask how you would program a robot to respond to a wall or a chair. Remind students that you cannot simply say "Step over chair" unless you know there is a chair, and you will not always know there is a chair. It might be helpful to translate the task into instructions like:

- while there is a path ahead
  - walk forward
  - if there is a chair, step over it
- sit down

Tell students they will be using **conditionals** to solve this problem on Code.org. Give the definition of:

- **Condition**: A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored.
- **Conditionals**: Statements that only run under certain conditions.

Open up a discussion of when you might use a conditional in your code.

Main Activity (30 min)

Course F Online Puzzles - Website

The patterns in these puzzles may not be obvious to every student. We recommend that you play through these levels beforehand to best understand any problem areas for your class. Also, watching and using the techniques from **Pair Programming - Student Video** may be helpful for your class.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a conditional? Why would you program a conditional?
- Give an example of using a conditional during your day? (ex.**If** I am hungry, I eat some food;**While** I am walking across the street, I keep an eye out for cars)

Extended Learning

While We Play

Gather the class for some fun outside or in a gym with a ball! This could be done in a circle or as a team in a court

Rules:

- **While** the ball is in play, we must all be ready to hit it
- **If** the ball is hit to you, you must keep it in the air
- **If** you hit the ball once, you cannot hit it again (only one touch per person per turn, no double hitting)
- **If** the ball goes out of bound, every student must fall to the ground dramatically. The last kid to fall has to retrieve the ball.

At the end of the first round, ask the students if they can identify the conditionals in the game. Can they come up with others they might want in the game?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 12: Functions with Minecraft

Functions | Minecraft

Overview

Students will begin to understand how functions can be helpful in this fun and interactive Minecraft adventure!

Purpose

Students will discover the versatility of programming by practicing functions in different environments. Here, students will recognize reusable patterns and be able to incorporate named blocks to call pre-defined functions.

Agenda

Warm Up (10 min)

Introduction

Bridging Activity - Functions (15 min)

Unplugged Activity Using Some Blockly
Preview of Online Puzzles

Main Activity (30 min)

Course F Online Puzzles - Website

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Use functions to simplify complex programs.
- Use pre-determined functions to complete commonly repeated tasks.

Preparation

□ Play through **Course F Online Puzzles - Website** to find any potential problem areas for your class.

□ Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

□ Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Function** - A named group of programming instructions. Functions are reusable abstractions that reduce the complexity of writing and maintaining programs.

Teaching Guide

Warm Up (10 min)

Introduction

Help the class understand that functions are simply a chunk of code that has a name. Once defined, you can use that name over and over in your program to tell the computer to run the chunk of code that you assigned to it.

Bridging Activity - Functions (15 min)

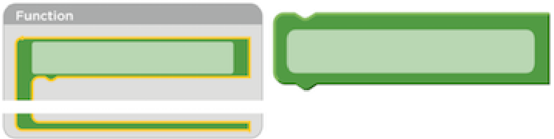
This activity will help bring the unplugged concepts from "Functions Unplugged: Songwriting" into the online world that the students are moving into. Choose **one** of the following to do with your class:

Unplugged Activity Using Some Blockly

Pick a song to play that the students enjoy and print out the lyrics. You can use the same song from "Functions Unplugged: Songwriting." Break your class into groups or pairs. Pass out the printed out lyrics (including the repeated chorus) and the basic function blocks from **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives** to each group or pair of students. See lesson tip for details.

Lesson Tip

Function blocks:



The block to the left is a function declaration, a block that students will name and use to fill in the function. The block to the right is a function call, a block that makes the function code run. Students will need multiple of the function call blocks.

Ask the students to cross out any part of the song that can be made into a function (the chorus is a good example) and put it into the function blocks provided. Students should fill in the function declaration with a function name and the words of the repeated lyrics. Once the function declaration is done, ask the students to fill in the function calls and place them on top of the crossed out lyrics.

Once every group or pair is done, ask the class where they put their functions and why. Did everyone make the same function? How often is the function repeated?

Preview of Online Puzzles

Pull up a puzzle from **Course F Online Puzzles - Website** . We recommend puzzle 9 of this lesson. As a class, work through the puzzle without using functions. Once you have gotten the solution, display it on a white board or overhead. Ask the class to point to the repeated code. Ask the class how they would simplify the program. Why can you not just use a loop?

On the white board or overhead, rewrite the program without the repeated code, but leaving one line space. In that/those line space(s), call a function. Off to the side, declare the function like the left example block in the lesson tip. Ask the class what they think the code will do now.

Open up a discussion with the class on why functions could be useful in programming. Invite students to discuss the difference between functions and loops.

Main Activity (30 min)

Course F Online Puzzles - Website

We recommend providing paper and pencils for students to write (or draw) out ideas. Also, if students are having trouble recognizing patterns, have them work with a partner on the harder puzzles.

Wrap Up (15 min)

Journaling

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

- What was today's lesson about?
- How do you feel about today's lesson?
- What did your functions do in the programs you wrote today? How did that help you?
- When should you use a function instead of a loop?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 13: The Power of Words

Common Sense Education | Cyberbullying

Overview

Students consider that while they are enjoying their favorite websites they may encounter messages from other kids that can make them feel angry, hurt, sad, or fearful. They explore ways to handle cyberbullying and how to respond in the face of upsetting language online.

Students discuss all the ways they use technology for communication and explore the similarities and differences between in-person and online communication. Students then brainstorm ways to respond to cyberbullying.

Purpose

This lesson will provide students with the tools that they need to handle cyberbullying if they are ever in the situation of having someone negatively responds to their online postings.

Students may not ever have the misfortune of experiencing cyberbullying, but they should understand what it is so that they can spot it online. Students will learn how to identify cyberbullying and what steps they should take to make it stop. This may become helpful in later puzzles when students have the opportunity to share their work.

Agenda

Warm Up (5 min)

- **Introduction**

Main Activity (35 min)

- **What's the Problem?**
- **Crossing the Line**
- **Talk and Take Action**

Wrap Up (15 min)

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

Assessment (10 min)

View on Code Studio

Objectives

Students will be able to:

- Empathize with those who have received mean and hurtful messages.
- Judge what it means to cross the line from harmless to harmful communication online.
- Generate solutions for dealing with cyberbullying.

Preparation

□ Preview the **Common Sense Education - Power of Words - Teacher Prep Guide** and prepare to show it to your class.

□ Print out the **Words Can Hurt Handout** from **Common Sense Education - Power of Words - Teacher Prep Guide** (page 7) for each group of four.

□ Print out the **Talk and Take Action Handout** from **Common Sense Education - Power of Words - Teacher Prep Guide** (page 6) for each student.

□ Print out the assessment on page 8-9 of **Common Sense Education - Power of Words - Teacher Prep Guide**.

□ Obtain colored pencils and a string the length of the classroom.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Common Sense Education - Power of Words** - Teacher Prep Guide
- **The Power of Words** - Lesson Video
- **Common Sense Education** - Website
- **CSF Digital Citizenship** - Resource List

For the Students

- **Feeling Faces** - Emotion Images
- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Cyberbully** - Using technology tools to deliberately upset someone else.

Teaching Guide

Warm Up (5 min)

Introduction

Draw a series of expressive faces on the board. View**Feeling Faces - Emotion Images** for examples.

Invite the students to suggest emotions that match each face's expression. With every suggestion, write the emotion next to the feeling face. Answers will vary.

Tell students that not everyone will react to a particular situation in the same way, but just because a reaction is different from our own, doesn't mean we should discount others' feelings.

Explain to students they are going to watch a video about how words, whether typed or spoken, can impact how someone else feels.

Show students **The Power of Words - Lesson Video** .

Ask:

- Who has heard of the saying, "Sticks and stones may break my bones, but words will never hurt me"?
- What did Guts mean in his text that sometimes words can hurt?
    - Words are powerful. Sometimes it is hard to ignore what someone is saying when it's a mean name. Names**can** make you feel sad or hurt.

Remind students to keep Leg's question in the back of their mind during this lesson**How do you treat others online?**

Main Activity (35 min)

What's the Problem?

Organize students into groups of four and have each group pick a person to record their ideas.

Distribute the **Words Can Hurt Student Handout** . Have the groups of students read the scenario about Rani and Aruna receiving mean messages through a children's game website.

Have each group answer the questions, then have them share their responses with the class. Look for responses that show empahty for Rani and Aruna and acknowledge that the messages sent to them were mean and hurtful. Ask the students to read the 'Use Common Sense!' section on the **Words Can Hurt Student Handout** .

Invite students to share their own stories.

Ask:

- Have you seen mean messages sent to you or others online? Tell us about it, but do not use real names.

Divide students into pairs.

Invite one partner to write the phrase "You're weird" on a piece of paper, then hand it to their partner. Tell them that they just received this text.

Ask:

- What are the reasons the person might have texted "You're weird"?
    - They're continuing an inside joke; the first person did something silly at an earlier time; a group of kids is teasing the kid; the person who sent the text really does think the person is weird but is afraid to say it to his or her face.
- How did the partner feel about being called weird?
    - Possibly like the other person was kidding around, but maybe that the person was teasing or being hurtful.

Tell one person from each pair to say to the other person, "You're weird," with a smile on their face.

Ask:

- Why might you feel differently if you could see the person?
  - People give non-verbal cues through facial expressions and body language.

Crossing the Line

Place the piece of string across the length of the classroom. Ask students to stand on one side of the line. Then ask them to imagine that they are online and somebody has sent them a message, which you will read to them. Tell the students to stay where they are if they think the message is okay; to cross over the line if they think the message is not okay; or to stand on the line if they think the message is in between.

Read:

- You are my friend.
- You are an idiot.
- I'm having a party and you're not invited.
- I like your new haircut.
- You are ugly.
- Thanks for the advice. Next time, will you tell me in person rather than through text?
- Did you finish your homework?
- Why is it taking you so long to finish it?
- You are such a freak.

Review with the students that kids like to go online and use cell phones to email, chat, watch videos, send messages, play games, and do homework. But sometimes the language can get mean or scary. Messages that make people feel bad cross the line. Sometimes that meanness is unintentional, but when people use tools such as the internet and cell phones to deliberately upset someone else over and over, that's cyberbullying.

Talk and Take Action

Have students return to their seats.

Discuss how easy is it to feel angry or upset when somebody sends you a mean or scary message online.

Define:

- **Cyberbullying**: Using technology tools such as the internet and cell phones to deliberately upset someone else.

Explain that cyberbullies deliberately try to make you feel that way, just like real-life bullies.

Discuss:

- Cooling down can be a good first step when you receive a mean message online. Taking a deep breath, counting backwards from 10, or pausing to think about what you will do next can give you time to think of the BEST way to handle the situation.
- Finding help or telling a trusted adult or friend can be a good way to take action. You shouldn't deal with the cyberbullying situation alone. The person you tell should be someone who wants to hear what you have to say and will help you work on a solution. Adults can be especially good because they often have the power to influence the situation or they can give you advice about what to do.
- Ignoring the person who is cyberbullying you can be very effective. Those who bully often like attention.
- Whatever you do, remember to keep a copy of your communication with the individual who is cyberbullying you. If you delete the communication, there is no proof of how the bully treated you if you need to show a trusted adult.

Distribute the **Talk and Take Action Student Handout** to each student. Encourage them to depict a cyberbullying scenario and possible solution. They can use pencils and paper to make the comics.

Wrap Up (15 min)

Flash Chat: What did we learn today?

You can use these questions to assess your students' understanding of the lesson objectives. You may want to ask students to reflect in writing on these questions in their **Think Spot Journal - Reflection Journal**.

Ask:

- Why is it a bad idea to send mean or scary messages online?
  - Because they can make the person who gets the message upset, angry, or scared.
- Why might there be more misunderstandings between people when they send online messages as opposed to a face-to-face discussion?
  - Online messages can be more confusing or scarier than face-to-face messages because there are no face-to-face cues to help you understand people's intentions.
- What can kids do when they get cyberbullying messages?
  1. Stay calm and take a deep breath
  2. Tell a friend or trusted adult who can help develop a plan to handle the situation
  3. Ignore the bully
  4. Keep a copy of the communication with the bully.

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is cyberbullying?
- Who are some people you can go to if you are ever bullied online or in person?

Assessment (10 min)

Print out the assessment from **Common Sense Education - Power of Words - Teacher Prep Guide** (page 8-9) and distribute it to the class. Give students enough time to complete the assessment, but make sure there is enough time to go over answers.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **NI** - Networks & the Internet

Lesson 14: Envelope Variables

Unplugged | Variable

Overview

Variables are used as placeholders for values such as numbers or words. Variables allow for a lot of freedom in programming. Instead of having to type out a phrase many times or remember an obscure number, computer scientists can use variables to reference them. This lesson helps to explain what variables are and how we can use them in many different ways. The idea of variables isn't an easy concept to grasp, so we recommend allowing plenty of time for discussion at the end of the lesson.

Purpose

Variables are very helpful in programming. Students will be introduced to this topic using envelopes to represent variables that have been given names. The value of the variable will be written on a card inside of an envelope. This lesson helps students understand how names can be a placeholder for values in the physical world, so that programming with variables will seem less confusing in the virtual world.

Agenda

Warm Up (10 min)

- **Vocabulary**
- **Introduction**

Main Activity (20 min)

- **Envelope Variables - Worksheet**

Wrap Up (10 min)

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

Assessment (10 min)

- **Envelope Variables - Assessment**

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Identify variables and determine their values.
- Define and call variables in the context of real-life activities.
- Create situations which require the use of variables.

Preparation

□Watch the **Envelope Variables - Teacher Video**.

□Obtain 6 or more blank envelopes for warm up plus some for the main activity.

□Print one **Envelope Variables - Worksheet** per student.

□Print one **Envelope Variables - Assessment** for each student.

□Provide students with envelopes, paper, pens & pencils.

□Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

> **Heads Up!** Please make a copy of any documents you plan to share with students.

For the Teacher

- **Variables in Envelopes** - Unplugged Video (**download**)
- **Envelope Variables** - Teacher Video
- **Envelope Variables** - Worksheet
- **Envelope Variables** - Worksheet Answer Key
- **Envelope Variables** - Assessment
- **Envelope Variables** - Assessment Answer Key

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Variable** - A placeholder for a piece of information that can change.

Teaching Guide

Warm Up (10 min)

Vocabulary

This lesson has one important word:

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

A placeholder for a piece of information that can change.

Introduction

Call four volunteers to the front of the room and line them up. Let the students know that you are going to write a poem for each of them.

On the board (or under your document camera) write the sentence for your first student (suppose it's Bill):

"My student Bill, standing proud
is a fine example for the crowd"

Encourage the students to clap at your abilities and thank Bill for volunteering. Allow Bill to sit down (or go to the back of the line) as you erase the board, then call the next volunteer (we'll say that she's called Annie).

"My student Annie, standing proud
is a fine example for the crowd"

Again, accepting applause, erase the board and invite the next volunteer.

"My student Jenny, standing proud
is a fine example for the crowd"

As you call the final volunteer, inquire as to whether everyone in the class would like a poem written about each of them. Maybe the everyone in the whole school? Goodness, that's going to take a while! Pose the question to your students:

"How could I do this more quickly?"

Your students will likely pick up on the fact that only one word is changing, and that word is simply a person's name. Help them see the location by circling Jenny's name on the board and writing "firstName" next to it.

"It would take a long time to write a poem for everyone in the school if I couldn't start until I knew who I was writing it about, wouldn't it?"

- How long do you think it would take to make a video game if they couldn't start until they knew your username?
- How expensive would video games be if they had to be created separately for each person?
- How do you think we can get around that?

By this time, it's quite likely that your class will come up with the idea of having a placeholder. With that, they're most of the way into understanding where this lesson goes.

- What would we call that placeholder?
    - We need to call it something that makes sense. We wouldn't want to call it "age" if it was a placeholder for their name, right?

Now, let's add some more volunteers. Give them each a piece of paper to write their name on, and have them tuck it inside individual envelopes labeled firstName.

This time, put the poem on the board with a blank space labeled "firstName" where the student's name will go.

- Have the first student in line (likely the last student from the previous example) pull their name from the envelope and that's what you'll write in the space.
- When you erase the board, only erase the portion with the last student's name in it.

- Call the next student to show their variable.
- Repeat as many times as is entertaining

Now it's time for the main activity.

Main Activity (20 min)

Envelope Variables - Worksheet

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

Directions:

- Divide students into groups of 2-4.
- Have students design (draw) a robot.
- After 10-15 minutes, request that the students fill their envelopes with important details about their robot such as its name, height, and purpose.
- Collect each group's envelopes, then bring them to the front of the room to share.
- Write on the board, "My robot's name is robotName, it is numUnitsTall tall, and it's purpose is purpose."
- Use the envelopes to fill the appropriate variable in the sentence, then ask each group to stand when they hear the sentence that describes their creation.

Wrap Up (10 min)

Flash Chat: What did we learn?

- What did we learn today?
- Can you think of anywhere that you have seen variables before?
- There is at least one variable at the top of most homework hand outs? Can you think of what it could be?
- Why do you think that professionals do not put spaces in variable names?
  - What would happen if there was a variable "eye" a variable "color" and a variable "eye color"?
- Variables can be used to store numbers, too.
  - Suppose I have envelopes labeled num1 and num2, then I write num1+num2?
  - What happens if the "num1" envelope contains the number 4 and "num2" contains the number 5?

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a variable?
- Why do you think variables are important in programming?

Assessment (10 min)

Envelope Variables - Assessment

Allow students enough time to finish this assessment. If you are willing to spare more time, go over the answers as a class.

Extended Learning

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

What's in the box?

- Draw boxes on a piece of paper with simple mathematical operators between them.
  - For instance [] + [] = []
- Have similar size squares with numbers between 1 & 20.
- Ask one student to come create a true equation, using the numbers provided.
- Once the student has finished (and the class verifies the equation) exchange one of the numbers with another one, then remove a second number entirely.
  - Tell the students that there is a hidden number in the empty box that makes that equation true again.
  - What number is in the box?
- Play this game over and over again until you can remove the number from any location and the students can figure out what it is supposed to be.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 15: Variables with Artist

Variable | Artist

Overview

In this lesson, students will explore the creation of repetitive designs using variables in the Artist environment. Students will learn how variables can be used to make code easier to write and easier to read, even when the values don't change at runtime.

Purpose

This stage teaches the most basic use for variables, as a constant that reoccurs frequently in a program.

Agenda

Warm Up (15 min)

Introduction

Bridging Activity - Variables (15 min)

Unplugged Activity Using Variables as Constant

Preview of Online Puzzles as a Class

Main Activity (30 min)

Course F Online Puzzles - Website

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Assign values to existing variables.
- Utilize variables in place of repetitive values inside of a program.

Preparation

□ Play through the **Course F Online Puzzles - Website** associated with this lesson to find any potential problem areas for your class.

□ Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

□ Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Variables** - Student Video (**download**)
- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Constant** - A variable used throughout a program that never changes value
- **Variable** - A placeholder for a piece of information that can change.

Teaching Guide

Warm Up (15 min)

Introduction

This is the first online lesson dealing with variables, so it might be a worthwhile exercise to review the "Envelope Variables" unplugged activity from last time, as well as the vocabulary that was introduced in that lesson.

Discussion:

- What is a variable? (A placeholder for a piece of information that can change.)
- When can a variable be helpful? (When you don't know what information is going to be used in a certain place until runtime, or when you have lots of places that one piece of information will be used, but that information might change someday.)

Ask the class when they could see a variable being helpful in programming. When would they NOT want to use a variable?

If the class seems interested, continue the discussion. Otherwise, move on to one of the bridging activities.

Bridging Activity - Variables (15 min)

This activity will help bring the unplugged concepts from "Envelope Variables" into the online world that the students are moving into.

Unplugged Activity Using Variables as Constant

Discuss: Remember our robot article? It used each variable only once...but what if I wanted to mention the robot's name several times?

Display: Write a paragraph on the board that refers back to the name of a specific robot several times. Ask the students what happens when you need to make the article about a different robot.

Think/Pair: Ask students to work together to see if they can come up with an idea to make changing the article for each robot easier.

Share: Work with students until you eventually get to the place where you have defined a variable called**robotName** somewhere before the paragraph, then set **robotName** equal to the robot that you are writing the article about. Replace all specific naming instances for the previous robot with the variable **robotName**.

Preview of Online Puzzles as a Class

Demo: Display a puzzle for the class. We recommend the 6th puzzle. Go over the code with the students to make sure they understand what's happening before they help you convert the code to use variables. Can they think of something that might happen that would make them really glad that they used variables instead of hardcoded numbers?

Transition: Now it's time for your students to move to their own machines and get started!

Main Activity (30 min)

Course F Online Puzzles - Website

Notice that this stage first covers the idea of a variable as a constant (a variable that you use in many places, but it does not change.) This might be something that students find helpful as they're creating their own projects.

Watch out for puzzle #6. It is the first time that students will be expected to set a variable on their own. This can be tricky if they don't have a true grasp on the concept. If they're having trouble, send them back to the prediction level (#5) and have them explain to their partners why the answer ended up what it was. Once both partners are convinced, let them continue back to puzzle #6.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- What is a variable? Why is it helpful in programming?
- How well do you think you understand variables? (Answer on a scale from 1-5 or with an emoticon.) If you're having troubles, can you put into words what you don't understand?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

COURSE F 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 CODE

Lesson 16: Changing Variables with Bee

Variables | Bee

Overview

This lesson will help illustrate how variables can make programs more powerful by allowing values to change while the code is running.

Purpose

You don't always know what a value is going to be before you begin your program. Sometimes, values change while your code is running. This lesson will illustrate how code with changing values can be helpful.

Agenda

Warm Up (15 min)

Introduction

Main Activity (30 min)

Course F Online Puzzles - Website

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Identify areas where they can use variables to modify quantities during runtime.
- Examine code to find places where variables can be substituted for specific values.

Preparation

□Play through the **Course F Online Puzzles - Website** associated with this lesson to find any potential problem areas for your class.
□Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
□Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Variable** - A placeholder for a piece of information that can change.

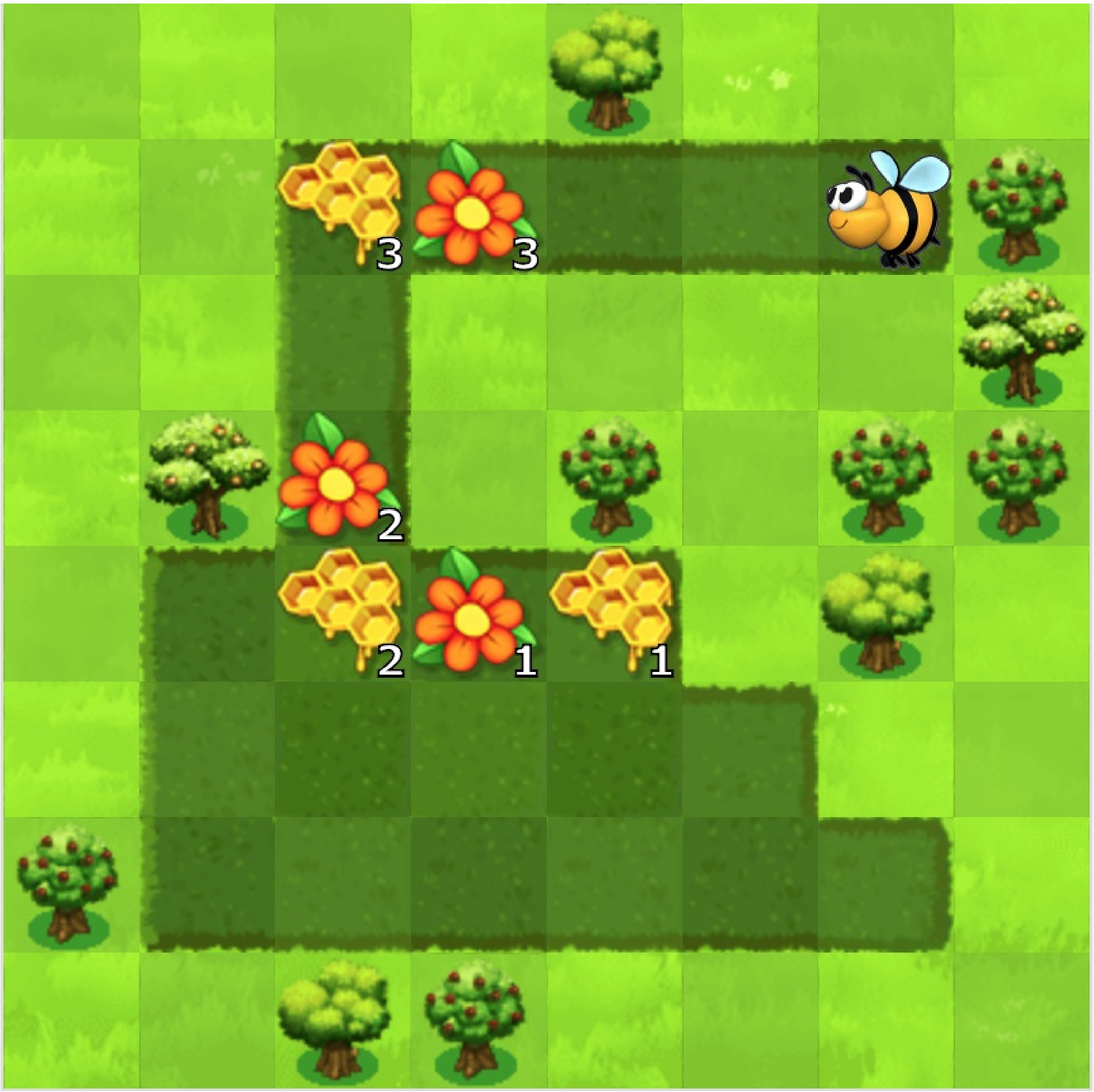
Teaching Guide

Warm Up (15 min)

Introduction

This series is a little different than what students have done in the past. Now, instead of simply assigning a value to a variable and running your code, you'll need to help students see how a variable can be modified during program runtime.

Display Show students the play area from one of the later puzzles.



There are several things to unpack here, so you might need to give your students a chance to look at it critically before you expect them to do anything with it.

Think/Pair: Suppose all of the flowers/honeycomb in this picture had the same amount of nectar/honey. How would you solve this puzzle?

Share: Let students share their ideas until you decide on one that the entire class is good with.

Discuss: Now, imagine that we didn't want to have to write a separate chunk of code for each set of flowers and honeycomb. How could we use a variable to have our loop do this for us?

(Eventually, you'll want to get to the place where you initialize a variable to the original value, then change it each time through the loop so that it's ready for the next time.)

Main Activity (30 min)

Course F Online Puzzles - Website

This set of puzzles takes some serious computational thinking skills. If you find that students are getting stuck, help them break down the puzzles into the individual pieces:

- What would it look like if the flowers/honeycomb all had the same amount of nectar/honey?
- What would it look like without the functions?
- Now how can you use a variable to get the quantities the way you want them?
- Now can you build it back up to use a function?

Hint: Puzzle 7 becomes much easier if students utilize the `while path ahead` loop instead of a variable.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- What are some ways you have used variables so far?
- What else do you think you can do with variables?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 17: Changing Variables with Artist

Variable | Artist

Overview

In this lesson, students will explore the creation of repetitive designs using variables in the Artist environment. Students will learn how variables can be used to make code easier to write and easier to read. After guided puzzles, students will end in a freeplay level to show what they have learned and create their own designs.

Purpose

Variables are essentially placeholders for values that might be unknown at the time that you run your program or for values that can change during the execution of a program. These are vital to creating dynamic code because they allow your program to change and grow based on any number of potential modifications. This stage reinforces the use of variables, using the most basic capabilities of setting and using them.

Agenda

Warm Up (5 min)

Introduction

Main Activity (20 min)

Course F Online Puzzles - Website

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Assign values to existing variables.
- Utilize variables in place of repetitive values inside of a program.
- Use variables to change values inside of a loop.

Preparation

□Play through the **Course F Online Puzzles - Website** associated with this lesson to find any potential problem areas for your class.

□Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

□Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Variables** - Student Video (**download**)
- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Variable** - A placeholder for a piece of information that can change.

Teaching Guide

Warm Up (5 min)

Introduction

It might be helpful to remind students of "Variables that Change in Bee," since variables will be used in a similar way here.

- How can we change the value of a variable inside of a loop?
- Do we have to change the value of a variable only by one each time?

Main Activity (20 min)

Course F Online Puzzles - Website

The latter half of this series is made up of freeplay puzzles. Students will have the opportunity to play with variables, shape, and color to create something unique.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- Have you tried mixing multiple variables into one program? What might that look like? When would it be helpful?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 18: For Loop Fun

Unplugged | For Loops

Overview

We know that loops allow us to do things over and over again, but now we're going to learn how to use loops that have extra structures built right in. These new structures will allow students to create code that is more powerful and dynamic.

Purpose

At this point, students have become masters of loops. Today, they will learn about another loop commonly used in programming. The `for` loop repeats commands a certain number of times, but also keeps track of the values it is iterating over. For example, a `for` loop that begins at 4, ends with 8, and has a step value of 1 will repeat 4 times, but the values 4, 5, 6, and 7 will also be captured for use elsewhere. Using this structure with variables can create some pretty fantastic programs. Today, students will simply be learning the basics of a `for` loop before diving into programming with them next time!

Agenda

Warm Up (20 min)

Vocabulary
For One and All

Main Activity (20 min)

For Loop Fun - Worksheet

Wrap Up (15 min)

Flash Chat: What did we learn?
Journaling

Assessment (5 min)

For Loop Fun - Assessment

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Determine starting value, stopping value, and stepping value for a `for` loop.
- Illustrate the counter values hit each time through a for loop during runtime.

Preparation

□Watch the **For Loop Fun - Teacher Video**.
□Watch the **For Loop Fun - Lesson in Action Video**.
□Print one **For Loop Fun - Worksheet** per group.
□Print one **For Loop Fun - Assessment** for each student.
□Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **For Loop Fun** - Unplugged Video (**download**)
- **For Loop Fun - Assessment Key** - Teacher Key (**download**)
- **For Loop Fun** - Teacher Video
- **For Loop Fun** - Lesson in Action Video
- **For Loop Fun** - Worksheet
- **For Loop Fun** - Worksheet Answer Key
- **For Loop Fun** - Assessment

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **For Loop** - Loops that have a predetermined beginning, end, and increment (step interval).

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

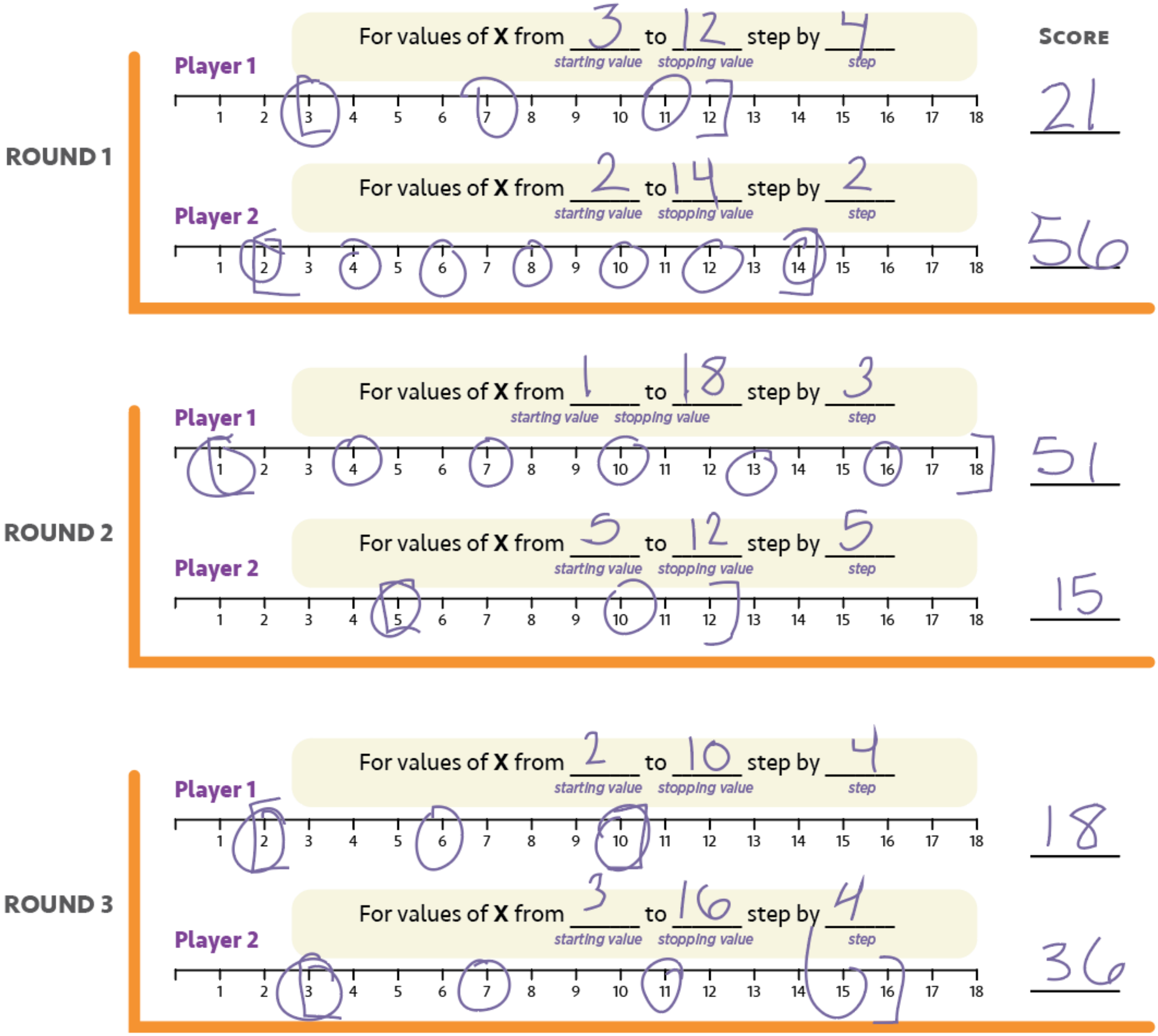
- **For Loop** - Say it with me: For-Loop

Loops that have a predetermined start, stop, and step value.

For One and All

- Point out that there are certain loops that happen very frequently, for example, loops where you need to keep track of how many times you have been through
  - Sometimes, you don't want to start with one
  - Sometimes, you don't want to count by ones
  - `for` loops give you a powerful way to keep a counter that starts when you want, ends when you want, and increases by whatever size step that you want

Here, you can jump right into a sample of the game (example in English)



ROUND 1

Player 1

For values of **X** from 3 to 12 step by 4
starting value stopping value step

21

Player 2

For values of **X** from 2 to 14 step by 2
starting value stopping value step

56

ROUND 2

Player 1

For values of **X** from 1 to 18 step by 3
starting value stopping value step

51

Player 2

For values of **X** from 5 to 12 step by 5
starting value stopping value step

15

ROUND 3

Player 1

For values of **X** from 2 to 10 step by 4
starting value stopping value step

18

Player 2

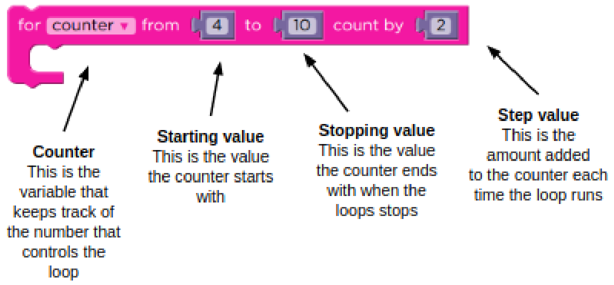
For values of **X** from 3 to 16 step by 4
starting value stopping value step

36

Main Activity (20 min)

For Loop Fun - Worksheet

Sometimes we want to repeat things a certain number of times, but we want to keep track of values as we do. This is where a for loop comes in handy. When you use a for loop, you know right from the start what your beginning value is, what your ending value is, and how much the value changes each time through the loop.



for counter from 4 to 10 count by 2

Counter
This is the variable that keeps track of the number that controls the loop

Starting value
This is the value the counter starts with

Stopping value
This is the value the counter ends with when the loops stops

Step value
This is the amount added to the counter each time the loop runs

for Loop block (in English)

Directions:

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.

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 + step){
circle currentXvalue;
add currentXvalue to roundScore;
}
```

- Divide students into pairs
- To start the round, each student rolls three times:
  - One die to determine the starting value of X
  - Three dice to determine the stopping value for X
  - One die to determine the stepping value of X each time through
- Use one of the provided number lines to trace the `for` loop that they've made
  - Start at the starting value of X
  - Count down the number line, circling the numbers at the rolled interval
  - Stop when you get to the predetermined stopping value
- Add all of the circled values to your score, then let the other player take a turn
- Best 2 out of 3 wins

Lesson Tip

If any of the values rolled are outside the bounds of the game (like rolling a start value of 6 but then rolling 2, 1, 2 for the end value), have the student re-roll everything.

Wrap Up (15 min)

Flash Chat: What did we learn?

- What would your interval need to be if you wanted to count from 4 to 13 by threes?
- What kinds of things do you think you could do with a for loop?
- Can you reproduce a normal loop using a for loop?
- What would you need to do?

Lesson Tip

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

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a `for` loop?
- Why would you use a `for` loop instead of a `repeat` loop or a `while` loop?

Assessment (5 min)

For Loop Fun - Assessment

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

Extended Learning

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

Run it Backward

- Try this activity again, but this time have the start number be selected using three dice, and the stop number with only one. Make sure to have a negative increment!

Hop Scotch

- Using chalk, draw a hop scotch diagram outside on the blacktop
    - Number the squares from bottom to top
    - Have students give each other a start square, stop square, and how many at a time they need to jump
    - When the jumper is done, have them write down the loop they just performed
    - Start adding additional activities to be done at each square, this will add complexity to the written portion, as well

Standards Alignment

CSTA K-12 Computer Science Standards

▶ **AP** - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes,**contact us.**

Lesson 19: For Loops with Bee

For Loop | Bee

Overview

Featuring Bee, this lesson focuses on `for` loops and using an incrementing variable to solve more complicated puzzles. Students will begin by reviewing loops from previous lessons, then they'll walk through an introduction to `for` loops so they can more effectively solve complicated problems.

Purpose

Today's concept, `for` loops, are a very important topic in computer science. Not only are they widely used, the process of learning `for` loops enhances the learning of other important concepts (such as variables and parameters.) Students will have plenty of practice critically thinking through problems by determining the starting, ending, and stepping values for each `for` loop. This concept uses plenty of math as well, so feel free to pair it with a math lesson for an even deeper learning experience.

Agenda

Warm Up (15 min)

Introduction

Bridging Activity - For Loops (15 min)

Unplugged Activity Using Paper Blocks
Previewing Online Puzzles as a Class

Main Activity (30 min)

Course F Online Puzzles - Website

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Determine starting value, stopping value, and stepping value for a `for` loop.
- Recognize when to use a `for` loop and when to use other loops such as `repeat` and `while` loops.

Preparation

☐Play through the **Course F Online Puzzles - Website** associated with this lesson to find any potential problem areas for your class.
☐Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
☐Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **Think Spot Journal** - Reflection Journal

Vocabulary

- **For Loop** - Loops that have a predetermined beginning, end, and increment (step interval).

Teaching Guide

Warm Up (15 min)

Introduction

Remind students of the work they did in "For Loop Fun". Open a discussion about what they learned, why they think it might be useful, and if they had any fun. Here are some discussion starters.

- What did you learn in "For Loop Fun"?
- What are the three main components of a for loop?
  - starting value, step interval, ending value
- Why do you think a for loop might be helpful in programming?
  - Many students might not know an answer to this. Let them hypothesize, but don't dwell on this question for too long.
- Did you have fun learning about for loops? Why or why not?
- Are you excited to use for loops in online puzzles?

Bridging Activity - For Loops (15 min)

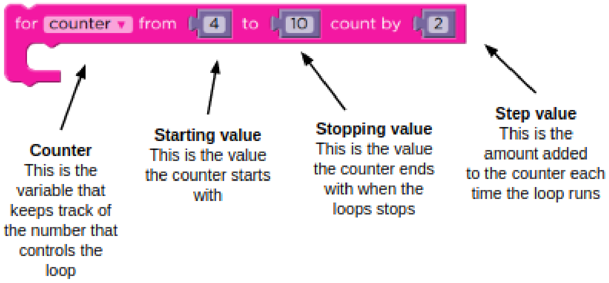
This activity will help bring the unplugged concepts from "For Loop Fun" into the online world that the students are moving into. Choose **one** of the following to do with your class:

Unplugged Activity Using Paper Blocks

Split up the students of your class into pairs. Ideally have the pairs be the same from when your class did "For Loop Fun". Print out a for loop from **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives** for each pair of students. Pass out one die to each pair. Have the partners take turns rolling the die to obtain the following values:

- One roll for the starting value
- Four rolls for the ending value
- One roll for the step value

Have each pair fill in the for loop with the appropriate values in the correct spot.


for counter from 4 to 10 count by 2
Counter This is the variable that keeps track of the number that controls the loop
Starting value This is the value the counter starts with
Stopping value This is the value the counter ends with when the loops stops
Step value This is the amount added to the counter each time the loop runs

Using a basic number line, like the one used in "For Loop Fun", have the students mark the beginning, ending, and middle values that this for loop will touch. When everyone is done, see who got the most points by totaling the starting, middle, and ending numbers of each pair.

Previewing Online Puzzles as a Class

Display a puzzle from the **Course F Online Puzzles - Website** associated with this lesson. We recommend puzzle #4 because it displays a potential solution and asks the user to evaluate it.

Using a number line similar to the ones used in "For Loop Fun", mark the start and ending values of the given for loop (if you aren't using puzzle #4, you will need to come up with a potential solution first). With the class's help, circle the values between the start and end that the `for` loop will touch. If you are working on puzzle #4, ask the class what they think the answer is to the question, given what they found with the number line.

Main Activity (30 min)

Course F Online Puzzles - Website

Some students may have a hard time differentiating between `repeat` loops and `for` loops. We recommend having scratch paper out for students to make guesses on values like the start, stop, and step. Implementing pair programming amongst the class might also be helpful for your students.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- How is a `for` loop different from a `repeat` loop?
- Why do you think `for` loops could be useful?

Standards Alignment

CSTA K-12 Computer Science Standards

- ▸ **AP** - Algorithms & Programming

Lesson 20: For Loops with Artist

For Loop | Artist

Overview

In this lesson, students continue to practice for loops, but this time with Artist. Students will complete puzzles combining the ideas of variables, loops, and for loops to create complex designs. At the end, they will have a chance to create their own art in a freeplay level.

Purpose

Creativity and critical thinking come together beautifully in this lesson. Students will continue their practice with for loops and variables while they create jaw-dropping images. This lesson inspires a creative mind while teaching core concepts to computer science.

Agenda

Warm Up (15 min)

Introduction

Main Activity (30 min)

Course F Online Puzzles - Website

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Use `for` loops to change loop several times with different values.
- Recognize when to use a `for` loop and when to use other loops such as `repeat` and `while` loops.

Preparation

□Play through the **Course F Online Puzzles - Website** associated with this lesson to find and potential problem areas for your class.

□Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

□Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

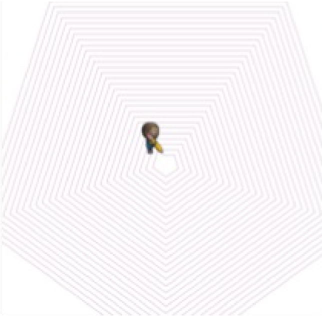
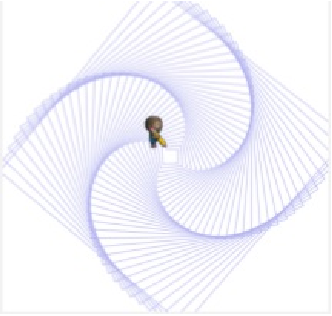
- **For Loop** - Loops that have a predetermined beginning, end, and increment (step interval).

Teaching Guide

Warm Up (15 min)

Introduction

On a board displayed to the entire class, draw (or display via projector) one of the final projects from the**Course F Online Puzzles - Website** associated with this lesson. We recommend one of the following:

Ask the class how a computer might draw the drawing you displayed.

After a few predictions have been said, reply with `for` loops of course!

Tell the students they will soon be learning how to create these fine drawings using`for` loops and variables.

Main Activity (30 min)

Course F Online Puzzles - Website

These puzzles are super fun, but it may be helpful for students to have protractors and scratch paper to see these designs made in the physical form. If that isn't an option in your class, try to get the students to trace on the computer screen with their fingers.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw one of the designs you made today. What was the code needed to create it?
- What are some designs you would like to create? How do you think`for` loops or variables could help create those?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 21: Learning Sprite Lab

Event | Sprite Lab

Overview

In this lesson, students will learn about the two concepts at the heart of Sprite Lab: sprites and behaviors. Sprites are characters or objects on the screen that students can move, change, and manipulate. Behaviors are actions that sprites will take continuously until they are stopped.

Purpose

This lesson is designed to introduce students to the core vocabulary of Sprite Lab, and allow them to apply concepts they learned in other environments to this tool. By creating a fish tank, students will begin to form an understanding of the programming model of this tool, and explore ways they can use it to express themselves.

Agenda

Warm Up (10 min)

Introduction
Review of "Personal and Private Information"

Main Activity (20 min)

Course F Online Puzzles - Website

Wrap Up (15 min)

Journaling

View on Code Studio

Objectives

Students will be able to:

- Define “sprite” as a character or object on the screen that can be moved and changed.
- Create a new sprite and choose its appearance.
- Create a new sprite and choose its appearance.

Preparation

□ Play through the **Course F Online Puzzles - Website** associated with this lesson to find any potential problem areas for your class.

□ Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

□ Make sure ever student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Sprite Lab Pilot Doc** Make a Copy ▾
- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Behavior** - An action that a sprite performs continuously until it’s told to stop.
- **Sprite** - A graphic character on the screen with properties that describe its location, movement, and look.

Teaching Guide

Warm Up (10 min)

Introduction

Today students will learn how to work with sprites in Sprite Lab.

Display: Pull up a previous puzzle from Code.org, ideally one containing a "main character" like Scrat from Ice Age or one of the Angry Birds.

Discuss: Let the students know that this character on the screen is a "sprite." It is a graphic that is controlled by a program. In this lesson, students will have the opportunity to choose their own sprites to control.

Display: Begin by showing Puzzle 1 to your students.

Think/Pair: Ask them to predict what will happen when the code is run, and to discuss with their neighbors. Run the code, and discuss the outcome.

Now is a great time to introduce the lesson vocabulary.

Review of "Personal and Private Information"

Remind students of information that is safe to share online and information that is strictly private.

SAFE - Personal Information	UNSAFE - Private Information
Your favorite food Your opinion *(though it shoud be done respectfully)* First name *(with permission)*	Mother's maiden name Social Security number Your date of birth Parents' credit card information Phone number

Discuss other examples of the two categories above.

Main Activity (20 min)

Goal: Today, students will be creating their own Fish Tank. They'll begin by learning how to put some sprites on the screen, then they will make them move. Finally, they'll customize their fish tank to add whatever creatures and objects they want.

Course F Online Puzzles - Website

Transition: Move students to their machines. Encourage students to follow the instructions for each puzzle. Help them realize that this is a creative activity, intended to help them learn Sprite Lab. It is not an assessment activity of any sort.

Reminder: If puzzles are sharable, remind the students to only share their work with their close friends or family. For more information watch or show the class **Pause and Think Online - Video**.

Teaching Tip

Encourage students with questions/challenges to start by asking their partner. Unanswered questions can be escalated to a nearby group, who might already know the solution. Have students describe the problem that they're seeing:

- What is it supposed to do?
- What does it do?
- What does that tell you?

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- How did it feel to make a scene that was more creative?
- Was it difficult to finish a lesson where there was no clear "right" and "wrong"?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 22: Alien Dance Party

Event | Sprite Lab

Overview

This lesson features Sprite Lab, a platform where students can create their own alien dance party with interactions between characters and user input. Students will work with events to create game controls.

Purpose

Students will use events to make characters move around the screen, make noises, and change backgrounds based on user input. This lesson offers a great introduction to events in programming and even gives a chance to show creativity! At the end of the puzzle sequence, students will be presented with the opportunity to share their projects.

Agenda

Warm Up (15 min)

- **Introduction**
- **Review of "Personal and Private Information"**

Main Activity (30 min)

- **Course F Online Puzzles - Website**

Wrap Up (15 min)

- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Identify actions that correlate to input events.
- Create an animated, interactive game using sequence and events.

Preparation

□Play through the **Course F Online Puzzles - Website** associated with this lesson to find any potential problem areas for your class.

□Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

□Make sure ever student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Sprite Lab Pilot Doc** Make a Copy ⌄
- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Introduction

Today students will visit **events** in programming.

Demo: Ask the students to raise their hands in the air.

What you did was declare an event. When you say "raise your hands in the air" the students responded by raising their hands. In coding, you would declare this by saying something like "when I say 'raise your hands,' you raise your hands".

You can also think of cities as declaring events. There are laws that say "when there is a green light, cars move through the intersection".

Discuss: Ask the students why they think this is an event.

Today, students will play in Play Lab, but the events they will be working on will be more like the video games they are used to playing. Events will take the form of actions, such as clicking the screen or two characters running into each other.

Display: Begin by showing Puzzle 1 to your students.

Think/Pair: Ask them to predict what will happen when the code is run, and to discuss with their neighbors. Run the code, and discuss the outcome.

Review of "Personal and Private Information"

Remind students of information that is safe to share online and information that is strictly private.

SAFE - Personal Information	UNSAFE - Private Information
Your favorite food Your opinion *(though it shoud be done respectfully)* First name *(with permission)*	Mother's maiden name Social Security number Your date of birth Parents' credit card information Phone number

Discuss other examples of the two categories above.

Main Activity (30 min)

Goal: Today, students will be creating their own alien dance party! They'll begin by reviewing how to put sprites on the screen, then they will assign them behaviors and learn to change those behaviors when an event is initiated.

Course F Online Puzzles - Website

Transition: Move students to their machines. Encourage students to follow the instructions for each puzzle. Help them realize that this is a creative activity, intended to help them learn Sprite Lab. It is not an assessment activity of any sort.

Reminder: If puzzles are sharable, remind the students to only share their work with their close friends or family. For more information watch or show the class **Pause and Think Online - Video** .

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- How did it feel to have control over what your characters were able to do?
- Did you change the program in any way to make it feel more like your own?

Teaching Tip

Encourage students with questions/challenges to start by asking their partner. Unanswered questions can be escalated to a nearby group, who might already know the solution. Have students describe the problem that they’re seeing:

- What is it supposed to do?
- What does it do?
- What does that tell you?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 23: Pet Giraffe

Event | Sprite Lab

Overview

Here, students will use Sprite Lab to play with sprites and their properties. Students will use events, behaviors, and custom code to create their very own pet giraffe that gets hungry, playful, and even filthy!

Purpose

Students will use events to make characters move around the screen, change size, and change colors based on user input. This lesson offers a great introduction to events in programming and even gives a chance to show creativity!

Agenda

Warm Up (15 min)

- **Introduction**
- **Review of "Personal and Private Information"**

Main Activity (30 min)

- **Course F Online Puzzles - Website**

Wrap Up (15 min)

- **Journaling**

View on Code Studio

Objectives

Students will be able to:

- Identify actions that correlate to input events.
- Create an animated, interactive game using sequence and events.

Preparation

□ Play through the **Course F Online Puzzles - Website** associated with this lesson to find any potential problem areas for your class.

□ Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

□ Make sure ever student has a**Think Spot Journal - Reflection Journal**.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Online Puzzles** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations

For the Students

- **Think Spot Journal** - Reflection Journal

Teaching Guide

Warm Up (15 min)

Introduction

Today students will revisit **events** in programming.

Review: Ask students questions about the Alien Dance Party.

- Do you remember what an event is?
- Can you name any of the events that you used to make the aliens dance? What do they do?
  - when clicked
  - when sprite1 touches sprite2
  - when arrow pressed

Display: Begin by showing Puzzle 1 to your students.

Think/Pair: Ask them to predict what will happen when the code is run, and to discuss with their neighbors. Run the code, and discuss the outcome.

Review of "Personal and Private Information"

Remind students of information that is safe to share online and information that is strictly private.

SAFE - Personal Information	**UNSAFE - Private Information**
Your favorite food Your opinion *(though it shoud be done respectfully)* First name *(with permission)*	Mother's maiden name Social Security number Your date of birth Parents' credit card information Phone number

Discuss other examples of the two categories above.

Main Activity (30 min)

Goal: Today, students will be creating their own pet giraffe! They'll begin by reviewing how to put sprites on the screen, then they will assign events that cause actions and behaviors upon interaction.

Course F Online Puzzles - Website

Transition: Move students to their machines. Encourage students to follow the instructions for each puzzle. Help them realize that this is a creative activity, intended to help them learn Sprite Lab. It is not an assessment activity of any sort.

Reminder: If puzzles are sharable, remind the students to only share their work with their close friends or family. For more information watch or show the class **Pause and Think Online - Video** .

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What other options would you like to be able to have your pet do?

Teaching Tip

Encourage students with questions/challenges to start by asking their partner. Unanswered questions can be escalated to a nearby group, who might already know the solution. Have students describe the problem that they're seeing:

- What is it supposed to do?
- What does it do?
- What does that tell you?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 24: Explore Project Ideas

Project | Define | Prepare | Try | Revise | Reflect

Overview

The next five lessons provide an opportunity for students to put their coding skills to use in a capstone project. This project will help individuals gain experience with coding and produce an exemplar to share with peers and loved ones. This is intended to be a multi-lesson or multi-week project where students spend time brainstorming, learning about the design process, building, and then presenting their final work.

In the "Explore" stage, students will play around with pre-built Artist and Sprite Lab programs for inspiration. Next, students will learn about the design process and how to implement it in their own projects. They will then be given the space to create their own project in Artist, Sprite Lab, or any other interface that you are comfortable providing. (This is likely the longest stage of the project.) Students will then revise their code after testing and peer review. Finally, students will be able to present their finished work to their classmates.

Purpose

Exploring project ideas is meant to inspire students with realistic and entertaining ideas for their culminating projects.

Agenda

Day 1 - Explore Project Ideas (45 min)

- **Example Projects**

Day 2 - The Design Process (45 min)

- **Define and Prepare**

Day 3 - Build Your Project (45 min)

- **Try**

Day 4 (Recommended for 5th Grade) - Revise Your Project (45 min)

- **Reflect and Try Again**

Day 5 & 6 - Present Your Project (45 min each)

- **Presentations**

Extension Activity

View on Code Studio

Objectives

Students will be able to:

- Learn to plan in advance for an ongoing assignment.
- Be able to explain how system limitations can affect project design.
- Describe how compromise can help keep a project on track and inspire creativity.

Preparation

- □ Play through the online **Course F Project Examples** to get an idea of the strengths, weaknesses, and limitations of the tool.
- □ Decide whether or not you will have your students do the section on **Revisions** (recommended for Course F).
- □ Print one copy of the **Design Process - Teacher Prep Guide** for each student.
- □ Modify the **CS Fundamentals Final Project - Rubric** to fit your class goals and print out a copy for each student.
- □ Modify the **Final Project Design - Worksheet** to fit your class and print one packet for each student.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Course F Project Examples**
- **Design Process** - Teacher Prep Guide

  Make a Copy ▾
- **Final Project Design** - Worksheet

  Make a Copy ▾
- **CS Fundamentals Final Project** - Rubric
- **72 Creative Ways for Your Students to Show What They Know** - Website

- **Google Slides**

Vocabulary

- **Define** - Figure out the details of the problems that you are trying to solve
- **Prepare** - Research, plan, and acquire materials for the activity you are about to do
- **Reflect** - Carefully think back on something with the intention of improving the outcome in the future
- **Try** - Attempt to do something

Teaching Guide

Day 1 - Explore Project Ideas (45 min)

Example Projects

Goal: This part of the process is an exploration. Students will sit down with a stage full of example projects to remix and learn. Not only will this give students an idea of what is possible, it will also help them see the limitations of the tool.

Give students a day to play with and remix the projects found in**Course F Project Examples** . Have them use their journals (or notebook paper) to keep track of thoughts and ideas as they go.

This activity should be done in the same pairs/groups that will be working on projects together over the next several lessons.

Make sure your class understands that they will be spending the next several weeks working with projects of their own, so they should pay close attention to how these programs were written, as well as the concepts that they use.

Day 2 - The Design Process (45 min)

Define and Prepare

Goal: Students will come up with a project and plan their strategy for programming that project in a single day. Students should have a project sketch and a description by the time the day is done.

Preparing Students for the Process:

The most important responsibility you have in kicking off this segment is to help your class understand the scope of this project. Students should be clear about the various expectations over the coming weeks so that they can prepare for their review and presentations appropriately.

To help your class manage this multi-stage undertaking, they should be given both the**Final Project Design - Worksheet** and the **CS Fundamentals Final Project - Rubric** on the first day of planning. Students will then be able to follow the rubric each step of the way to predict what their project grade will be in the end.

The Final Project Design Worksheet will provide a place for students to capture relevant thoughts and processes as they go, so they are more prepared for their reviews and presentations in the end.

As the teacher, you should download a copy of the documents and decide which elements are important to you. Be sure to edit or remove anything that you do not intend to draw student focus.

Define and Prepare:

Now that the class has their Final Project Design Worksheet in hand, they should start filling out the questions under **Day 1**.

Students will likely need to refer back to their notes from playing with the example projects, especially if they don't have access to online Artist or Play Lab project levels while they plan.

Lesson Tip:

Save 5 minutes or so at the end of the day to have students trade Final Project Design Worksheets to look at each other's work. This will help make sure that nothing is omitted or overlooked.

Students should focus on defining and planning their project during Day 1, and not cross over into building until their ideas have been written up and/or drawn out.

If students get stuck, help them work through ideas by asking questions and recalling examples, rather than offering solutions.

Day 3 - Build Your Project (45 min)

Try

Goal: Students will use this day to build an initial version of their project.

Equipped with their Final Project Design Worksheet, students should head to the computers to start bringing their projects to life.

This process will come complete with plenty of trial and error. Projects are likely to become truncated versions of the original scope (if not morphed altogether). Remind students that this kind of compromise is common in software design, but they need to be sure to document the reasons for the changes in their product.

Don’t let the class forget to fill out their Final Project Design Worksheets as they go. It might be helpful to suggest that pairs/groups take a worksheet break to begin discussing these questions about halfway through their lab time. Alternatively, the navigator can keep their eyes open for pertinent answers while the driver codes.

Be sure that each team member has their own Final Project Design Worksheet, as there are questions about each student’s own individual thoughts and behaviors that need to get captured along the way.

Day 4 (Recommended for 5th Grade) - Revise Your Project (45 min)

Reflect and Try Again

Goal: Students will work with another group to give and receive feedback in an effort to make each other’s projects stronger.

Reflect:

For reflections, have each group pair up with another group to try each other’s projects. After about 10 minutes, have the groups discuss the questions in the Final Project Design Worksheet.

Encourage students to ask the questions on the FPDW and write down feedback provided by their reviewing teams so that they can refer back to it later. This portion should take approximately 15 more minutes.

Try Again:

With their new reflections in hand, students can head back to their machines to make a handful of edits. With just 10 minutes left, they will likely have to select only the most important feedback to incorporate.

Lesson Tip:

Teachers should avoid assigning the final bit of project work as homework unless they are certain that students both live within a close proximity to one another **and** have internet access at home.

Day 5 & 6 - Present Your Project (45 min each)

Presentations

Goal: Students will create and present their projects in an approved manner (written, oral, or using multimedia).

Create:

Ideally, you will have class time available to give students to work on their presentations. This will allow them to incorporate rich multimedia components, like **Google Slides**. For other presentation ideas, visit**72 Creative Ways for Your Students to Show What They Know - Website**.

Lesson Tip:

If you are looking for a section of this series to assign as homework, this is it! Projects do not have to be presented in electronic form, so this is a great offline option.

Encourage students to include all of the information from Section J of the Final Project Design Worksheet into their presentation, as well as two or more questions from Section K.

Present

Students should showcase their apps first, then they can discuss the questions that they covered in their presentations.

It can be very helpful to have students sign up for a specific order in which to give their presentations, so that they are able to enjoy the demonstrations of their classmates without worrying about whether they will be called on next.

Extension Activity

If your students are already comfortable with coding concepts, try having them create their projects in another platform, like **Scratch** or **Alice**.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 25: The Design Process

Project

Overview

Over the course of five lessons, students will be building up to building a project of their own design using either Sprite Lab or Artist as their programming environment. In this portion of the project, students will learn about the design process and how to implement it in their own projects. The lesson guide overviewing all five stages of the process can be found in the beginning of the project process, **here**.

Purpose

Students may be ready to jump straight into building their projects, but this lesson will help shape their ideas into plans. This structure will keep the dreamers grounded and illuminate a path for those feeling left in the dark.

Agenda

Day 2 - The Design Process (45 min)

Define and Prepare

View on Code Studio

Objectives

Students will be able to:

- Shape ideas into reasonable goals and plans.
- Recognize any potential obstacles such as time constraints or bugs.

Teaching Guide

Day 2 - The Design Process (45 min)

Define and Prepare

Students will come up with a project and plan their strategy for programming that project in a single day. Students should have a project sketch and a description by the time the day is done.

Preparing Students for the Process:

The most important responsibility you have in kicking off this segment is to help your class understand the scope of this project. Students should be clear about the various expectations over the coming weeks so that they can prepare for their presentations appropriately.

To help your class manage this multi-stage undertaking, they should be given both the**Final Project Design - Worksheet** and the **CS Fundamentals Final Project - Rubric** on the first day of planning. Students will then be able to follow the rubric each step of the way to predict what their project grade will be in the end.

The **Final Project Design - Worksheet** will provide a place for students to capture relevant thoughts and processes as they go, so they are more prepared for their presentations in the end.

As the teacher, you should decide which elements of these documents are important to you and be sure to edit or remove anything that you do not intend to draw student focus.

Define and Prepare:

Now that the class has their**Final Project Design - Worksheet** in hand, they should start filling out the questions under **Day 1**.

Students will likely need to refer back to their notes from playing with the example projects, especially if they don't have access to online Artist or Play Lab project levels while they plan.

Lesson Tip

Save 5 minutes or so at the end of the day to have students trade their **Final Project Design - Worksheet** to look at each other's work. This will help make sure that nothing is omitted or overlooked.

Students should focus on defining and planning their project during Day 1, and not cross over into building until their ideas have been written up and/or drawn out.

If students get stuck, help them work through ideas by asking questions and recalling examples, rather than offering solutions.

Standards Alignment

CSTA K-12 Computer Science Standards

- **AP** - Algorithms & Programming

Lesson 26: Build Your Project

Project

Overview

Over the course of five lessons, students will be building up to building a project of their own design using either Sprite Lab or Artist as their programming environment. Now the students will be given their own space to create their project with either Artist or Sprite Lab. This will be the longest portion of the project. The lesson guide overviewing all five stages of the process can be found in the beginning of the project process, **here**.

Purpose

This lesson provides students with ample time to build and revise their projects. The trial and error inevitably involved in this lesson will teach problem solving and persistence.

Agenda

Day 3 - Build Your Project (45 min)

Try

View on Code Studio

Objectives

Students will be able to:

- Use the planned design as a blueprint for creation.
- Overcome obstacles such as time constraints or bugs.

Teaching Guide

Day 3 - Build Your Project (45 min)

Try

Students will use this day to build an initial version of their project.

Equipped with their **Final Project Design - Worksheet** , students should head to the computers to start bringing their projects to life.

This process will come complete with plenty of trial and error. Projects are likely to become truncated versions of the original scope (if not morphed altogether). Remind students that this kind of compromise is common in software design, but they need to be sure to document the reasons for the changes in their product.

Don't let the class forget to fill out their **Final Project Design - Worksheet** as they go. It might be helpful to suggest that pairs/groups take a worksheet break to begin discussing these questions about halfway through their lab time. Alternatively, the navigator can keep their eyes open for pertinent answers while the driver codes.

Be sure that each team member has their own Final Project Design Worksheet, as there are questions about each student's own individual thoughts and behaviors that need to get captured along the way.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▸ **AP** - Algorithms & Programming

Lesson 27: Revise Your Project

Project

Overview

Over the course of five lessons, students will be building up to building a project of their own design using either Sprite Lab or Artist as their programming environment. Now that the projects are built, students are given the opportunity to get feedback from peers and revise their projects. The lesson guide overviewing all five stages of the process can be found in the beginning of the project process, **here**.

Purpose

This lesson helps students take a step back and view their project from a new perspective. Here, students will be able to decide if they have reached their goals. If they haven't, this lesson gives them time and space to complete the project.

Agenda

Day 4 - Revise Your Project (45 min)

Reflect and Try Again

View on Code Studio

Objectives

Students will be able to:

- Determine if the criteria set in a rubric has been met with their current project.
- Draft and implement plans to resolve any issues in their code.

Teaching Guide

Day 4 - Revise Your Project (45 min)

Reflect and Try Again

Goal: Students will work with another group to give and receive feedback in an effort to make each other’s projects stronger.

Reflect:

For reflections, have each group pair up with another group to try each other’s projects. After about 10 minutes, have the groups discuss the questions in the Final Project Design Worksheet.

Encourage students to ask the questions on the FPDW and write down feedback provided by their reviewing teams so that they can refer back to it later. This portion should take approximately 15 more minutes.

Try Again:

With their new reflections in hand, students can head back to their machines to make a handful of edits. With just 10 minutes left, they will likely have to select only the most important feedback to incorporate.

Lesson Tip:

Teachers should avoid assigning the final bit of project work as homework unless they are certain that students both live within a close proximity to one another **and** have internet access at home.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▸ **AP** - Algorithms & Programming

Lesson 28: Present Your Project

Project

Overview

Over the course of five lessons, students will be building up to building a project of their own design using either Sprite Lab or Artist as their programming environment. Finally, students will be able to present their finished work to their peers or share with their loved ones with a special link. The lesson guide overviewing all five stages of the process can be found in the beginning of the project process, **here**.

Purpose

At this point, students have worked very hard on their projects, so this lesson is meant to offer a space for the students to share their projects. This lesson will build a supportive community where students will build their own confidence and feel connected to their hardworking peers.

Agenda

Day 5 & 6 - Present Your Project (45 min each)

Presentations

View on Code Studio

Objectives

Students will be able to:

- Clearly indicate where each criteria point from the rubric is satisfied in the code for the finished culminating project.
- Articulate the design process and how it helped shape the finished culminating project.

Teaching Guide

Day 5 & 6 - Present Your Project (45 min each)

Presentations

Students will create and present their projects in an approved manner (written, oral, or using multimedia).

Create:

Ideally, you will have class time available to give students to work on their presentations. This will allow them to incorporate rich multimedia components, like **Google Slides**. For other presentation ideas, visit**72 Creative Ways for Your Students to Show What They Know - Website**.

Encourage students to include all of the information from Section J of the Final Project Design Worksheet into their presentation, as well as two or more questions from Section K.

Lesson Tip:

If you are looking for a section of this series to assign as homework, this is it! Projects do not have to be presented in electronic form, so this is a great offline option. Other ways to present projects (both online and offline) include:

- Report
- Blog post
- Online
- In front of the class with a poster

Present:

Students should showcase their apps first, then they can discuss the questions that they covered in their presentations.

It can be very helpful to have students sign up for a specific order in which to give their presentations, so that they are able to enjoy the demonstrations of their classmates without worrying about whether they will be called on next.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **AP** - Algorithms & Programming