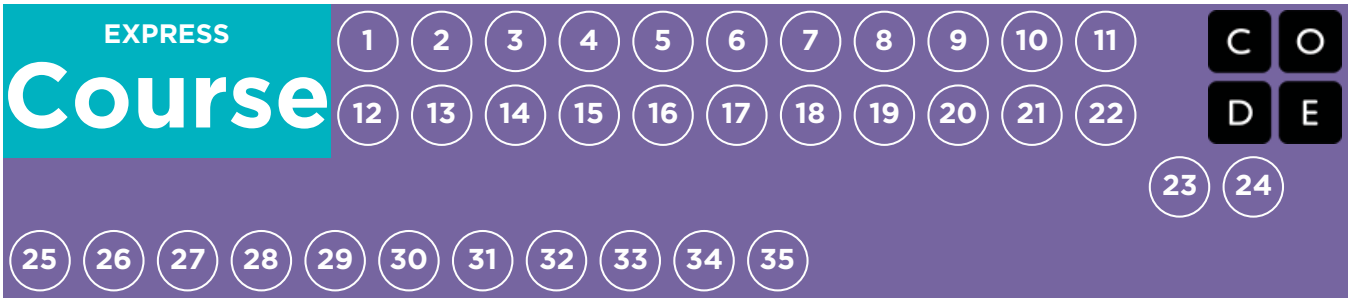


Express Course

Learn the basics of computer science and internet safety. At the end of the course, create your very own game or story you can share.

Teacher Links: Teacher Videos Playlist

Lesson 1: Programming: Graph Paper Programming

Unplugged | Programming | Program

In this lesson, you will program your friend to draw pictures!

Lesson 2: Introduction to Online Puzzles

Sequencing | Debugging | Loop | Angry Bird | Maze | Artist

Lessons 1-9 are considered ramp-up lessons. If you feel that the first few activities are too simple for your class, feel free to pick and choose your favorites or skip to lesson #10.

Lesson 3: Building a Foundation

Unplugged | Persistence | Frustration

Lesson 4: Debugging in Scrat

Debugging | Bug | Scrat | Ice Age

Have you ever run into problems while coding? In this lesson, you will learn about the secrets of debugging. Debugging is the process of finding and fixing problems in your code.

Lesson 5: Loops: My Robotic Friends

Unplugged | Loop | Repeat

In this activity you will "program" your friends using loops to stack cups to make cool towers!

Lesson 6: Loops in Artist

Loop | Artist

These online puzzles will get you to draw some amazing designs using loops!

Lesson 7: Nested Loops

Nested Loops | Loops

Lesson 8: Nested Loops in Frozen

Loop | Nested Loop | Artist

Here you will be using nested loops to make cool projects to show to your family and friends.

Lesson 9: Beyond Programming: The Internet

Unplugged | Internet

Ever wondered how information travels across the internet? It's not magic! This lesson will teach you the basics of how the internet works.

Lesson 10: Digital Citizenship: Practicing Digital Citizenship

Common Sense Education | Unplugged

Some information is not safe to share online. This lesson will help you learn the difference between safe and private information.

Lesson 11: Digital Citizenship: Screen Out the Mean

Common Sense Education | Cyberbullying | Unplugged

This lesson will teach you about mean things that happen online and how to respond to them.

Lesson 12: Events: The Big Event

Unplugged | Events

This lesson will teach you about events, which are necessary for the video games you play!

Lesson 13: Events in Star Wars

Star Wars | Event

This lesson will guide you through creating a Star Wars video game.

Lesson 14: Events with Flappy

Event | Flappy

Here you will make a super cool video game with Flappy!

Lesson 15: Events in Bounce

Event | Bounce

Ever wish you could play video games in school? In this lesson, you will get to make your own!

Lesson 16: Conditionals: Conditionals with Cards

Conditionals | Unplugged

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

Lesson 17: While Loops in Farmer

While Loops | Loops | Farmer

Loops are so useful in coding. This lesson will teach you about a new kind of loop: while loops!

Lesson 18: Conditionals & Loops in Maze

Conditional | Loop | Maze | Angry Bird | Zombie

You can do some amazing things when you use conditionals and loops together!

Lesson 19: Conditionals in Minecraft

Conditional | Minecraft

Avoid the lava! Here you will learn about conditionals in the world of Minecraft.

Lesson 20: Conditionals & Loops in Farmer

Conditional | Loop | Farmer

It's not always clear when to use each conditional. This lesson will help you get practice deciding what to do.

Lesson 21: Variables: 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 22: Variables in Artist

Variable | Artist

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

Lesson 23: Variables in Play Lab

Variable | Play Lab

Now you will learn about making characters interact in a game using variables!

Lesson 24: For Loops: For Loop Fun

Unplugged | For Loops

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

Lesson 25: For Loops in Bee

For Loop | Bee

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

Lesson 26: For Loops in 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 27: Functions: Songwriting with Parameters

Unplugged | Function | Parameter

You just might release the next big hit single! In this lesson, you will be learning what parameters are and how they make some fantastic songs!

Lesson 28: Functions in Bee

Function | Bee

The bee needs your help again! Here you will be using functions to get nectar and make honey!

Lesson 29: Functions with Parameters in Artist

Function | Parameter | Artist

Get your programming fingers ready. In these puzzles you will make impressive drawings in Artist using functions with parameters.

Lesson 30: Functions with Parameters in Bee

Function | Parameter | Bee

You've had a little practice using functions with parameters. This lesson will continue your practice with Bee!

Lesson 31: 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 32: The Design Process

Project

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

Lesson 33: Build Your Project

Project

Finally you can start building your project!

Lesson 34: 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 35: Present Your Project

Project

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

EXPRESS Course

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 29 30 31 32 33 34 35

C O D E

Lesson 1: Programming: Graph Paper Programming

Unplugged | Programming | Program

Overview

By "programming" one another to draw pictures, students will begin to understand what coding is really about. The class will begin by having students instruct each other to color squares on graph paper in an effort to reproduce an existing picture. If there's time, the lesson can conclude with images that the students create themselves.

Purpose

The goal of this activity is to build critical thinking skills and excitement for the course.

By introducing basic concepts like **programming** and **algorithms** to the class in an unplugged activity, students who are intimidated by computers can still build a foundation of understanding on these topics. Programming and algorithms are essential to computer science. In this lesson, students will learn how to translate instructions into a program and recognize an algorithm.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Introduction to Graph Paper Programming**
- **Practice Together**

Main Activity (20 min)

- **Graph Paper Programming - Worksheet**

Wrap Up (15 min)

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

Assessment (10 min)

- **Graph Paper Programming - Assessment**

Extended Learning

Objectives

Students will be able to:

- Understand the difficulty of translating real problems into programs.
- Arrange sequential events into their logical order.Explain how ideas may feel clear and yet still be misinterpreted by a computer.
- Practice communicating ideas through codes and symbols.

Preparation

- [ ] Watch the **Graph Paper Programming - Teacher Video**.
- [ ] Watch the **Graph Paper Programming - Lesson in Action Video**.
- [ ] Print out one **Graph Paper Programming - Worksheet** for each group.
- [ ] Print one **Graph Paper Programming - Assessment** for each student.
- [ ] Supply each group with several drawing grids, paper, and pens/pencils.
- [ ] Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Graph Paper Programming** - Unplugged Video (**download**)
- **Graph Paper Programming** - Teacher Video

- **Graph Paper Programming** - Lesson in Action Video
- **Graph Paper Programming** - Worksheet
- **Graph Paper Programming** - Worksheet Answer Key
- **Graph Paper Programming** - Assessment
- **Graph Paper Programming** - Assessment Answer Key

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has two new and important words:

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

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

- **Program** - Say it with me: Pro-gram

An algorithm that has been coded into something that can be run by a machine

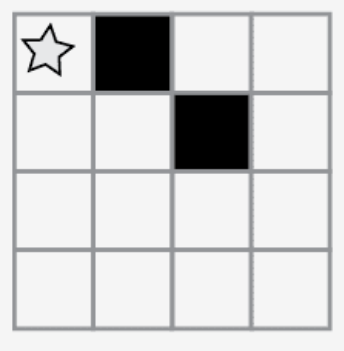
Introduction to Graph Paper Programming

In this activity, we are going to guide each other toward making drawings, without letting the other people in our group see the original image.

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

- Move One Square Right
- Move One Square Left
- Move One Square Up
- Move One Square Down
- Fill-In Square with color

For example, here's how we would write an algorithm to instruct a friend (who is pretending to be a drawing machine) to color their blank grid so that it looks like the image below:

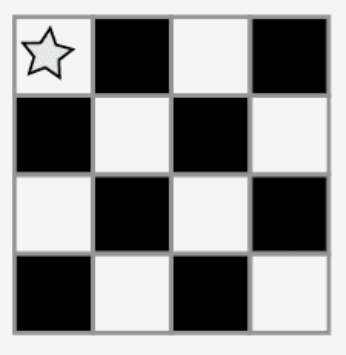


- Move One Square Right
- Fill In Square with Color
- Move One Square Right
- Move One Square Down
- Fill In Square with Color

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

Lesson Tip

Have the class imagine that your arm is an Automatic Realization Machine (ARM). The idea of "algorithms" and "programs" will be brought to life even further if students feel like they're actually in control of your movements.



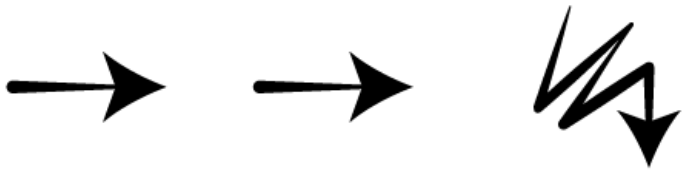
- Move One Square Right
- Fill In Square with Color
- Move One Square Right
- Move One Square Right
- Fill In Square with Color
- Move One Square Down
- Move One Square Left
- Fill In Square with Color
- Move One Square Left
- Move One Square Left
- Fill In Square with Color
- PLUS 12 MORE INSTRUCTIONS!

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

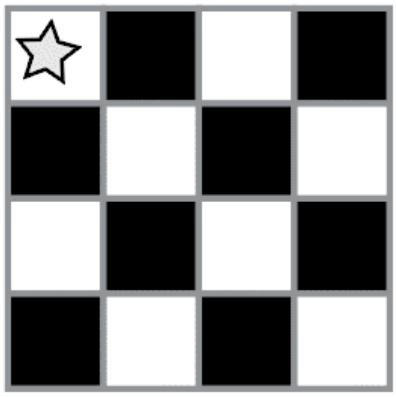
In this instance, the arrow symbols are the “program” code and the words are the “algorithm” piece. This means that we could write the algorithm:

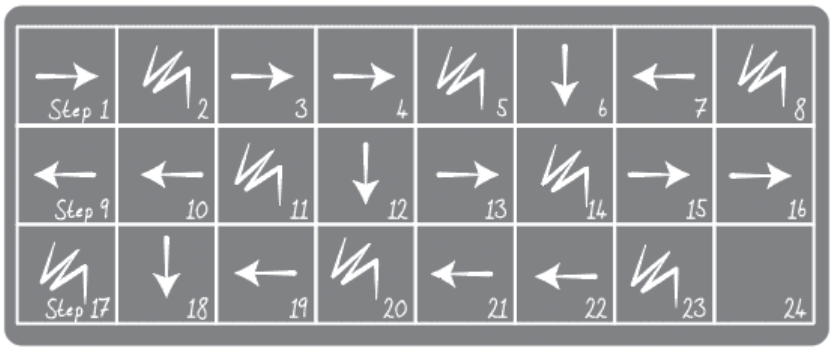
- “Move one square right, Move one square right, Fill-in square with color”

and that would correspond to the program:



Using arrows, we can redo the code from the previous image much more easily!
(Notice that we have written our program from left to right like you would read a book in English)





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

Practice Together

Start your class off in the world of programming by drawing or projecting the provided key onto the board.

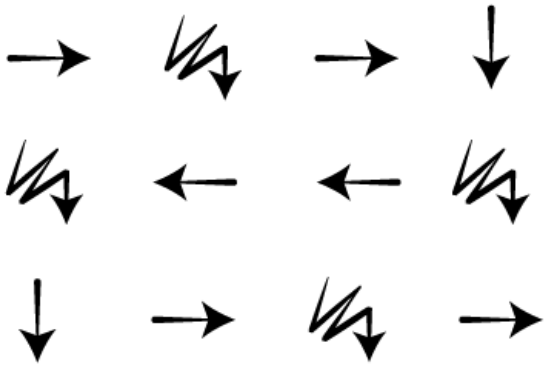
Select a simple drawing, such as this one to use as an example.

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

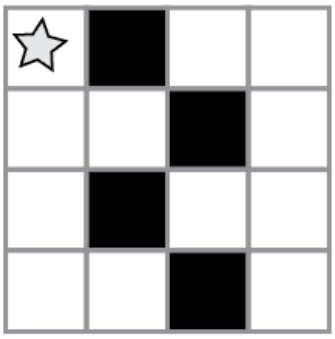
A sample algorithm:

"Move Right, Fill-In Square, Move Right, Move Down Fill-In Square, Move Left, Move Left, Fill-In Square Move Down, Move Right, Fill-In Square, Move Right"

Some of your class may notice that there is an unnecessary step, but hold them off until after the programming stage. Walk the class through translating the algorithm into the program:



The classroom may be buzzing with suggestions by this point. If the class gets the gist of the exercise, this is a good place to discuss alternate ways of filling out the same grid. If there is still confusion, save that piece for another day and work with another example.

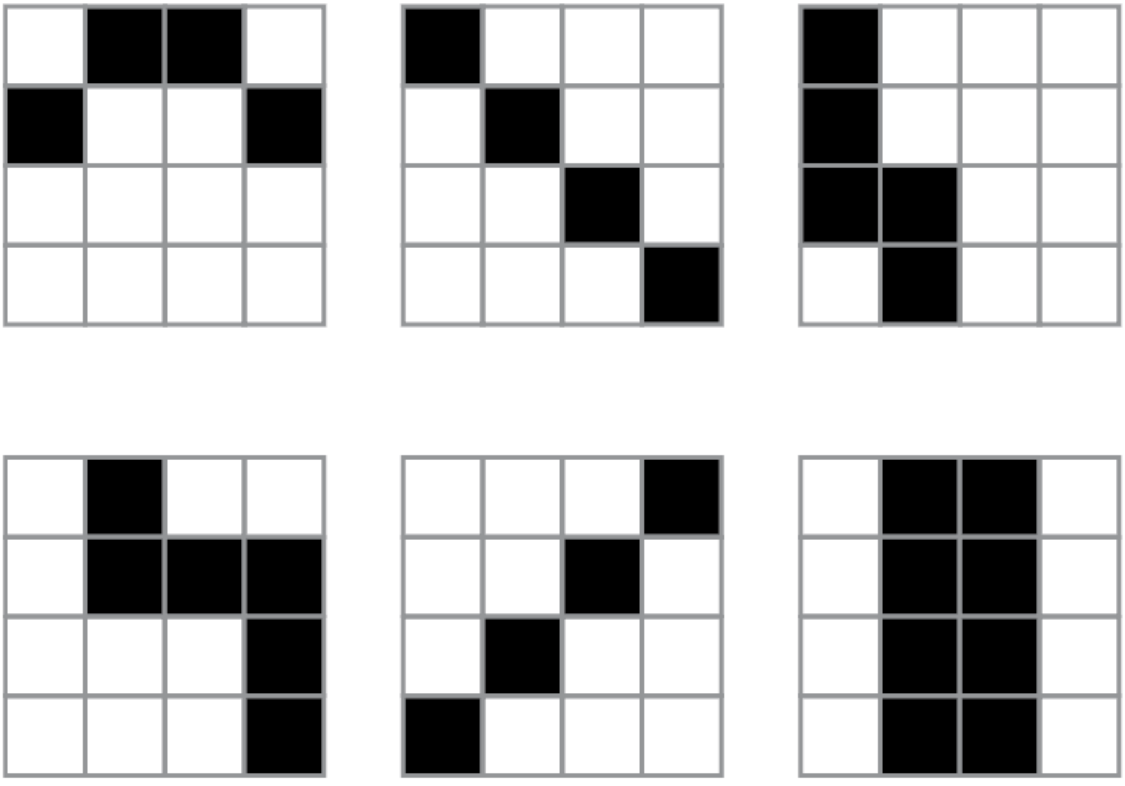


If the class can shout out the algorithm and define the correct symbols to use for each step, they're ready to move on. Depending on your class and their age, you can either try doing a more complicated grid together or skip straight to having them work in groups on their **Graph Paper Programming - Worksheet** .

Main Activity (20 min)

Graph Paper Programming - Worksheet

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



Wrap Up (15 min)

Flash Chat: What did we learn?

- What did we learn today?
- What if we used the same arrows, but replaced "Fill-In Square" with "Lay Brick"? What might we be able to do?
- What else could we program if we just changed what the arrows meant?

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw another image that you could code. Can you write the program to make this drawing?
- What are some other instructions that might come in handy for big or complicated images?

Assessment (10 min)

Graph Paper Programming - Assessment

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

Extended Learning

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

Better and Better

- Have your class try making up their own images.
- Can they figure out how to program the images that they create?

Class Challenge

- As the teacher, draw an image on a 5x5 grid.
- Can the class code that up along with you?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

EXPRESS Course

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 29 30 31 32 33 34 35

C O D E

Lesson 2: Introduction to Online Puzzles

Sequencing | Debugging | Loop | Angry Bird | Maze | Artist

Overview

In this progression, students will begin with an introduction (or review depending on the experience of your class) of Code.org's online workspace. Students will learn the basic functionality of the interface including the Run , Reset , and Step buttons. Dragging, deleting, and connecting Blockly blocks is also introduced in the beginning video. In the puzzles, students will practice their sequencing and debugging skills in Maze and Artist.

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 this "Ramp-Up Stage" for Course E. This can be used as either an introduction or a review to the Code.org interface and basic computer science concepts. This stage, along with the three that follow, cover all prerequisites needed to start Course E.

Agenda

Warm Up (15 min)

Introduction
Vocabulary

Bridging Activity - Programming (15 min)

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

Main Activity (30 min)

CSF Express Course - Website

Wrap Up (15 min)

Journaling

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 puzzles on **CSF Express Course - 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

- **CSF Express Course** - Website
- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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 (15 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 Activity - Programming (15 min)

This activity will help bring the unplugged concepts from "Graph Paper Programming" 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 "Graph Paper Programming", 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 the associated lesson on **CSF Express Course - 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 "Graph Paper Programming."

The class will not need to use the last arrow.

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

Main Activity (30 min)

CSF Express Course - 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:

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

- Utilize pair programming 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.

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 does it mean to "program"?
- Why is programming important?
- What's something about computers that you would like learn more about?

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

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.

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
- **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

EXPRESS Course 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 29 30 31 32 33 34 35 C O D E

Lesson 4: Debugging in Scrat

Debugging | Bug | Scrat | Ice Age

Overview

In this online activity, students will practice debugging with Scrat, from "Ice Age". Students will get to practice reading and editing code to fix puzzles with simple algorithms, loops and nested loops.

Purpose

The purpose of this lesson is to teach students that failure is normal when learning a new skill. Students will be given pre-written programs that do NOT work. They will be asked to fix these programs. This process, called "debugging", teaches students essential problem solving and critical thinking skills. These skills transfer over as students proceed to harder and harder programming projects.

Agenda

Warm Up (15 min)

- **Introduction**

Main Activity (30 min)

- **CSF Express Course - Website**

Wrap Up (15 min)

- **Journaling**

Objectives

Students will be able to:

- Read and comprehend given code.
- Identify a bug and the problems it causes in a program.
- Describe and implement a plan to debug a program.

Preparation

- [ ] Play through **CSF Express Course - 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

- **CSF Express Course** - Website
- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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.

Teaching Guide

Warm Up (15 min)

Introduction

One of the most important parts of learning to program is learning to debug. Ask the class if they have ever learned a new skill and faced failure.

For example:

- Learning to ride a bike and falling down
- Learning to bake and burning the food
- Learning to play a sport and not winning a game

Facing failure is very common when learning new things. Have students discuss past failures and how they overcame them.

In programming, computer scientists often run into "bugs" in their code.

- **Bug**: Part of a program that does not work correctly.

A bug can really mess up the program, so it's important to learn to "debug" your code.

- **Debug**: Finding and fixing problems in your algorithm or program.

Continue the conversation if you think your class needs more of an introduction, but leave time for one of the bridging activities.

Main Activity (30 min)

CSF Express Course - Website

It might be helpful for students to sit with their teams from the bridging activities. Every student should work on these puzzles individually or in pairs, but having a closely knit group to ask and answer questions with can help develop confidence and understanding with the subject matter.

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 bug? How do you know there is a bug in your program?
- What does it mean to debug code? How do you debug a program?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 5: Loops: My Robotic Friends

Unplugged | Loop | Repeat

Overview

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

Purpose

This lesson serves as an introduction to loops. **Loops** allow for students to simplify their code by grouping commands that need to be repeated. Students will develop critical thinking skills by noticing repetition in the movements of classmates and determining how many times their code needs to be looped.

Agenda

Warm Up (10 - 15 min)

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

Main Activity (15 - 20 min)

- **Looping with My Robotic Friends**

Wrap Up (8 min)

- **Journaling**

Extension Activities

Objectives

Students will be able to:

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

Preparation

- [ ] Watch the **My Loopy Robotic Friends - Teacher Video**.
- [ ] Print one **My Robotic Friends Loops - Teacher Prep Guide** for each group.
- [ ] Acquire up to 20 paper cups for each 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
- **My Robotic Friends** - Teacher Prep Guide
- **My Robotic Friends Loops** - Teacher Prep Guide
- **My Robotic Friends** - Teacher Video
- **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 - 15 min)

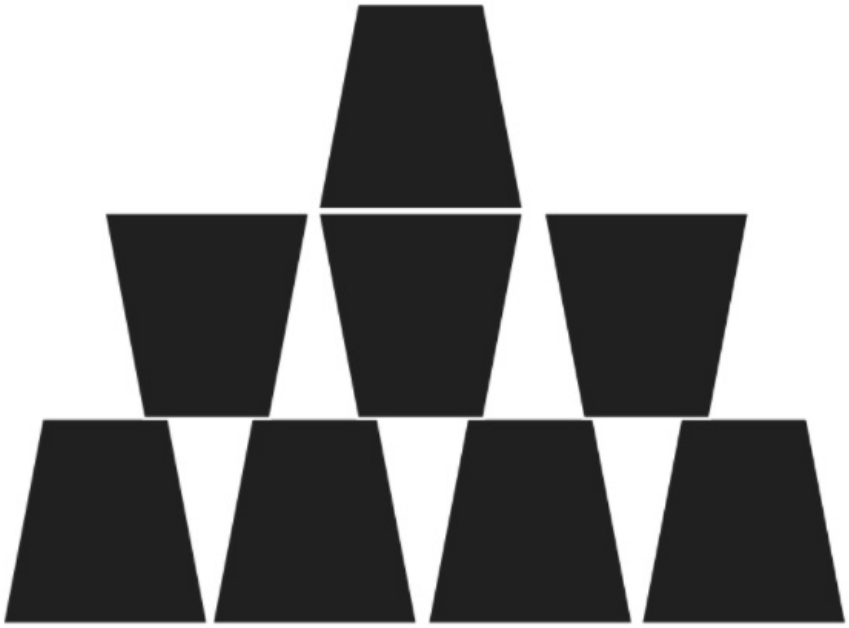
My Robotic Friends Review

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

Explain the rules of My Robotic Friends.

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

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



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

Now, bring up this image:

What is the reaction of the class?

Loops to the Rescue

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

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

Main Activity (15 - 20 min)

Looping with My Robotic Friends

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

Practice Makes Perfect

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

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

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

Looping with My Robotic Friends

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

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

Teaching Tips:

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

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

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

Wrap Up (8 min)

Journaling

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

Flash Chat:

Here are some possible topics:

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

Journal Prompts:

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

Extension Activities

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

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 6: Loops in Artist

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**

Bridging Activity - Loops (15 min)

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

Main Activity (30 min)

- **CSF Express Course - Website**

Wrap Up (15 min)

- **Journaling**

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 C 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

- **CSF Express Course** - Website
- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Repeat** - Do something again
- **while loop** - a programming construct used to repeat a set of commands (loop) as long as (while) a boolean condition is true.

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 dance with loops just like in "Getting Loopy"
- As a class, playing through a puzzle from the last lesson, "Loops in Maze"
- 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!

Bridging Activity - Loops (15 min)

This activity will help bring the unplugged concepts from "My Loopy 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

Select a pattern from **Stacking Cup Ideas - Manipulatives** from the My Robotic Friends unplugged activity and give students **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives** prefilled with the `repeat` loop and the `move___` commands. Next, have the students program a "robot" (a partner or the teacher) from their desks to get the correct stacking of the cups. Make sure that they understand that the blocks need to go from top to bottom and they all need to touch! Have the students pair share to check answers and resolve any questions or bugs that may come up.

Previewing Online Puzzles as a Class

Pull a puzzle from the corresponding online puzzles in **CSF Express Course - Website** . Using the **My Robotic Friends - Symbol Key** from the My Loopy Robotic Friends Unplugged Activity, have students draw out a pattern that they think the Artist will make. Ask the students to share. See how many other students had the same answer!

Main Activity (30 min)

CSF Express Course - 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

EXPRESS Course 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 29 30 31 32 33 34 35 CODE

Lesson 7: Nested Loops

Nested Loops | Loops

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)

CSF Express Course - Website

Wrap Up (15 min)

Journaling

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 **CSF Express Course - 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

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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)

CSF Express Course - 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

EXPRESS Course 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 29 30 31 32 33 34 35 C O D E

Lesson 8: Nested Loops in 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)

- **CSF Express Course - Website**

Wrap Up (15 min)

- **Journaling**

Objectives

Students will be able to:

- Define 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 D 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

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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)

CSF Express Course - 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 9: Beyond Programming: The Internet

Unplugged | Internet

Overview

Even though many people use the internet daily, not very many know how it works. In this lesson, students will pretend to flow through the internet, all the while learning about connections, URLs, IP Addresses, and the DNS.

Purpose

If you have been doing every lesson in this course, then each student in your classroom has used the internet...but how many know how it works? Learning more about the internet will help students develop a better understanding of its endless possibilities.

Agenda

Warm Up (20 min)

- **Vocabulary**
- **Getting the Message**

Main Activity (20 min)

- **The Internet**

Wrap Up (15 min)

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

Assessment (5 min)

- **Internet - Assessment**

Objectives

Students will be able to:

- Learn about the complexity of sending messages over the internet.
- Translate URLs into IP Addresses.

Preparation

- [ ] Watch the **Internet - Teacher Video**.
- [ ] Print enough **IP Address Cards and Delivery Type Cards - Manipulatives** for each group.
- [ ] Print one **Internet - Assessment** for each student.
- [ ] Access to the internet (such as **get-site-ip.com**).
- [ ] Make sure every student has a **Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Internet** - Teacher Video
- **IP Address Cards and Delivery Type Cards** - Manipulatives
- **Internet** - Assessment
- **Internet** - Assessment Answer Key

Vocabulary

- **DNS** - The service that translates URLs to IP addresses.
- **DSL/Cable** - A method of sending information using telephone or television

cables.

- **Fiber Optic Cable** - A connection that uses light to transmit information
- **Internet** - A group of computers and servers that are connected to each other.
- **IP Address** - A number assigned to any item that is connected to the Internet.
- **Packets** - Small chunks of information that have been carefully formed from larger chunks of information.
- **Servers** - Computers that exist only to provide things to others.
- **URL** - An easy-to-remember address for calling a web page (like www.code.org).
- **Wi-Fi** - A wireless method of sending information using radio waves.

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has several new and important words:

- **IP Address** - Say it with me: I-P Add-ress

A number assigned to any item that is connected to the Internet

- **DNS (Domain Name Service)** - Say it with me: D-N-S

The service that translates URLs to IP addresses

- **URL (Universal Resource Locator)** - Say it with me: U-R-L

An easy-to-remember address for calling a web page (like www.code.org)

- **Internet** - Say it with me: In-ter-net

A group of computers and servers that are connected to each other

- **Servers** - Say it with me: Ser-vers

Computers that exist only to provide things to others

- **Fiber Optic Cable** - Say it with me: Fye-ber Op-tic Cay-bl

A connection that uses light to transmit information

- **Wi-Fi** - Say it with me: Wye-Fye

A wireless method of sending information using radio waves

- **DSL/Cable** - Say it with me: D-S-L / Cay-bl

A method of sending information using telephone or television cables

- **Packets** - Say it with me: Pack-ets

Small chunks of information that have been carefully formed from larger chunks of information

> Lesson Tip
>
> A quick preview is all you need here. These words will all be explained as part of the lesson, so it would be far less confusing to do a brief intro to the words as a "see if you can spot these during the day" type of heads-up.

Getting the Message

- It's quite likely that your students are aware of what the internet is, but they may not really understand what the internet does.
  - Ask "What is the internet?"
  - Is the internet a public place or a private place?
  - (Truthfully, many people think it can be both, but it should be viewed as a public space no matter what settings you think you've mastered.)
  - How does information get from place to place?
- Let's say that I want to look at the webpage for Code.org. What do you suppose the process would be like for me to send a message to request that page?
  - What do I do as a user?
  - What do you think happens inside the internet?

> Lesson Tip
>
> There are some great YouTube videos on this subject that can make this lesson a little easier to understand. You can show them to the class in advance, or just watch them yourself. **Here is one of the most clear and entertaining versions**. (We recommend stopping the video at 2:59, if possible.)

Sending a message over the internet is a lot like sending a message through the mail...if every letter we sent required thousands of envelopes!

Every message we send through the internet gets chopped up and each piece is wrapped in its own version of an envelope. We call those "packets." Packets are specially formed chunks of information that are able to easily flow through any of the internet's channels.

Sometimes, a few of those packets will get lost, because the internet is a crazy place. In that case, the packets need to be resent, and the whole message has to get put on hold until they arrive.

Where do you think those packets are headed?

- Even if you're sending messages to another person, they first have to go to at least one "server."
  - A server is a special computer that is supposed to be always on and ready to send and receive information.
  - Every website has a server.
  - Even email goes through servers.

Servers don't have names like you and I do. They're actually addressed using numbers. These numbers are called IP addresses, and they look a little strange.

- For example: One of Code.org's IP addresses used to be 54.243.71.82
  - (Please be sure to check this out in advance. Most IP addresses change from time to time and they are then reused for other sites.)

There are many ways to reach the internet from your house, school, or place of business.

- You can connect directly using a cable (that might be DSL, Cable, or Fiber Optic)
- Or you can connect using radio waves over the air through Wi-Fi

Direct connections are most reliable, but they can be inconvenient.

- Can you figure out why?
  - (You have to be attached to a cable!)

Wi-Fi connections are super convenient, but the aren't always reliable.

- Can you figure out why not?
  - (Radio waves bounce all over the place and can get lost.)

So, if you're used to sending information to URLs (like **www.code.org**) and the servers actually have IP addresses for names (like 54.243.71.82) how does the Internet change from one to the other? That's what the DNS is for. The DNS (Domain Name Server) has tables that allow the system to go back and forth between URLs and IP addresses. If the Domain Name Servers ever stopped working, it would shut down the internet as we know it!

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

Lesson Tip

If you're thinking that this is a lot of text and it would be extremely boring to try to lecture this to a class full of elementary school kids, you're absolutely right! If you're unable to show a YouTube video in class to help explain it all, I highly recommend drawing pictures to explain each idea above, or choosing students as volunteers to act out what you describe while you're explaining. They're not expected to get every detail and definition at this point, only to gain exposure.

Pull out a piece of paper and draw a grid similar to that in the internet activity:

Sample of DNS Table:

#	URL	IP Address
1	code.org	54.243.71.82
2		
3		

#	URL	IP Address
4		
5		

First, we need to fill in this table.

- Survey the class for their favorite websites and write the URLs in the left column
- Use a site like **get-site-ip.com** to find the IP addresses for those sites and write them in the corresponding rows of the right column.

Now let's take this DNS Table and pretend to send messages through the internet!

Main Activity (20 min)

The Internet

Directions:

- Create your own DNS table, similar to what is shown above.
- Have the class help you fill in the blank spots in the table. Pick your favorite URLs and find their IP addresses using a site like **www.get-site-ip.com**.
- Divide into groups of 3 to 5.
- Assign each group an IP address from the newly created table, and assign each person in the group a position:
  - The Message Writer
  - The Internet
  - The Server (carries the IP address)
  - The Return Internet (optional)
  - The Message Receiver (optional)
- Each group will draw an **IP Address Cards and Delivery Type Cards - Manipulatives** to find out where their message is going and what their method of message delivery (Wi-Fi, Cable/DSL, or Fiber Optic Cable) will be.
- The Message Writer will craft a note to send to the server.
- The Internet will rip the message up into 4 small pieces called packets, then deliver each packet one at a time to the Server with the IP address that was drawn from the IP Address Card stack.
- The Server will make sure that the message arrives in order, then will send each packet off one at a time with the Return Internet (can be the same person or different person than the original Internet).
- The Return Internet will deliver each piece back to the Message Receiver (can be the same person or different person than the Message Writer) and put it back together.
- The Message Receiver will wait for all of the pieces to arrive, then read the message to be sure it arrived correctly!

Rules:

- The Internet must rip the message into exactly four packets.
- If the Internet drops a packet, they have to pick it up and go back to the start to deliver it again.
- The server has to wait for all of the message pieces to arrive before it can begin to send the message along.

Info:

- Wi-Fi: Convenient, but spotty. Wi-Fi doesn't require cables, but since the signal bounces all over the place, packets can get lost pretty easily.
  - Simulation: Internet must carry each packet on their shoulder (no hands).
- Cable/DSL: Fairly good at delivering messages, but you must be connected to a wire.
  - Simulation: Internet must carry each packet on the back of one hand and must keep the other hand touching a wall, desk, chair or the floor at all times.

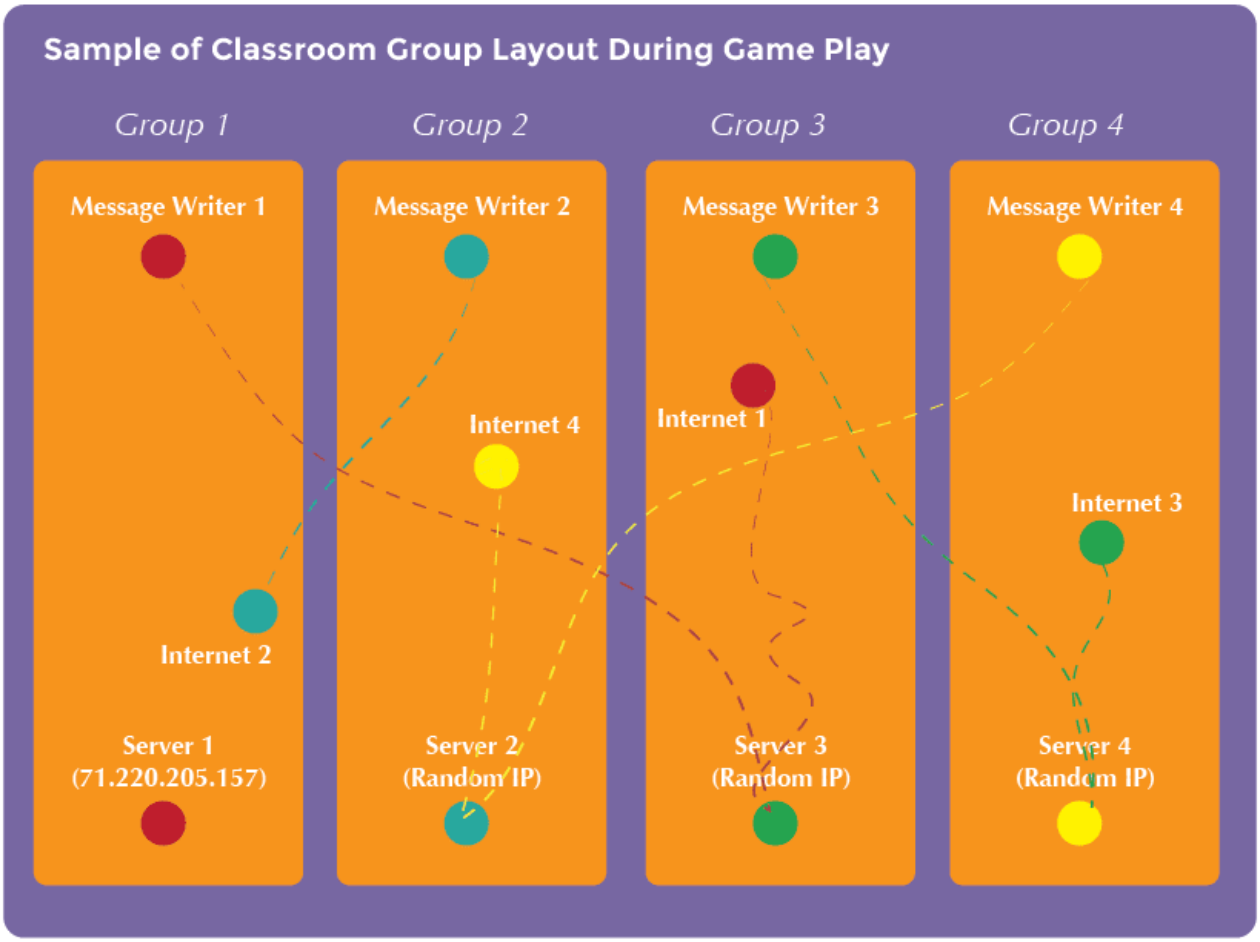
- Fiber Optic Cable: The best at delivering messages, but you must be connected to a wire.
  - Simulation: Internet can carry packets in hand, but must keep the other hand touching a wall, desk, chair or the floor at all times.

To play this game, you can have your groups cluster anywhere, but for the first time it can be less confusing to have groups play in a line.

Lesson Tip

If it feels like there are too many rules to explain outright, feel free to post them on the board and just explain the game as you go. You can play multiple rounds until the class really understands.

- Line up the "Servers" on one end of the room (holding their IP addresses). The Return Internet players can be over there as well (if you have that many people in each group).
- Have the everyone else line up across from their server at the other side of the room.
- The Message Senders will likely be sending their messages to a server other than their own, so the Internet players will likely cross over from group to group. It may look something like the diagram below (in English):



Wrap Up (15 min)

Flash Chat: What did we learn?

- What kind of connection would you rather have (Wi-Fi, DSL/Cable, or Fiber Optic)? Why?
- Why might it take your message a long time to get somewhere?

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What's something you learned about the internet today?
- Why is learning about the internet important?

Lesson Tip

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

Assessment (5 min)

Internet - Assessment

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

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **NI** - Networks & the Internet

EXPRESS Course

1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22
C O D E
23 24
25 26 27 28 29 30 31 32 33 34 35

Lesson 10: Digital Citizenship: Practicing Digital Citizenship

Common Sense Education | Unplugged

Overview

In collaboration with Common Sense Education, this lesson helps students learn to think critically about the user information that some websites request or require. Students learn the difference between private information and personal information, distinguishing what is safe and unsafe to share online.

Students will also explore what it means to be responsible and respectful to their offline and online communities as a step toward learning how to be good digital citizens.

Purpose

As students spend more time on computers, they should be aware that the internet is not always a safe space. In this lesson, students are taught what information is safe to share and what information should remain private. Students will create "superheros" and learn what it means to be a Digital Citizen on the internet.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **Personal vs. Private Online**

Main Activity (30 - 40 min)

- **Cubecraft Superhero Templates - Manipulatives**

Wrap Up (15 min)

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

Assessment (5 min)

- **Digital Citizenship - Assessment**

Extended Learning

Objectives

Students will be able to:

- Compare and contrast their responsibilities to their online and offline communities.
- Understand what type of information can put them at risk for identity theft and other scams.
- Reflect on the characteristics that make someone an upstanding citizen.Reflect on the characteristics that make someone an upstanding citizen.
- Devise resolutions to digital dilemmas.

Preparation

- [ ] Watch the **Digital Citizenship - Teacher Video**.
- [ ] Print out a good selection of male and female **Cubecraft Superhero Templates - Manipulatives** sheets for the whole class.
- [ ] Print one **Digital Citizenship - Assessment** for each student.

Links

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

For the Teacher

- **Digital Citizenship** - Teacher Video
- **Cubecraft Superhero Templates** - Manipulatives
- **Digital Citizenship** - Assessment
- **Digital Citizenship** - Assessment Answer Key
- **Common Sense Education** - Website

- **Think Spot Journal** - Reflection Journal

Vocabulary

- **Digital Citizen** - Someone who acts safely, responsibly, and respectfully online.

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important phrase:

Digital Citizen - Say it with me: Dih-jih-tal Sit-i-zen

Someone who acts safely, responsibly, and respectfully online

Personal vs. Private Online

- Ask "What types of information do you think are okay to share publicly online or on a profile that others will see?"
- What are some examples of websites where you must register in order to participate?
  - Write the names of the websites on the board.
- What information is required and why do you think it is required?
  - Information may be required to help distinguish one person from another.
  - The website may keep a record of who uses it.
- Explain that it's important to know that sharing some kinds of user information can put you and your family's privacy at risk.
- Point out that you do not have to fill out fields on websites if they are not required.
  - Required fields are usually marked by an asterisk (*) or are highlighted in red.
- Elementary school students should never register for sites that require private information without the approval and guidance of a parent or guardian.
- Here is an example of public versus private information:

Lesson Tip

If you have access to a computer, feel free to navigate to a site that might require this type of information, such as Gmail or Facebook.

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

- Explain that some people will actively try to get you to share this kind of information so that they can use it to take over your identity. Once a thief has taken someone's identity, he or she can use that person's name to get a driver's license or buy things, even if the person whose identity they stole isn't old enough to do these things!
  - It's often not until much later that people realize that their identity has been stolen. Identity thieves may also apply for credit cards in other people's names and run up big bills that they don't pay off. Let students know that identity thieves often target children and teens because they have a clean credit history and their parents are unlikely to be aware that someone is taking on their child's identity.

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

Main Activity (30 - 40 min)

common sense education®

Cubecraft Superhero Templates - Manipulatives

- Spiderman says "With great power comes great responsibility." This is also true when working or playing on the Internet.
- The things we read, see, and hear online can lead people to have all sorts of feelings (e.g., happy, hurt, excited, angry, curious).
  - What we do and say online can be powerful.
- The Internet allows us to learn about anything, talk to people at any time (no matter where they are in the world), and share our knowledge and creative projects with other people.
  - This also means that negative comments can spread very quickly to friends of all ages.
- CREATE a three-column chart with the terms “Safe,” “Responsible,” and “Respectful” written at the top of each column. Invite students to shout out words or phrases that describe how people can act safely, responsibly, and respectfully online, and then write them in the appropriate column.

Safe	Responsible	Respectful

Now, let's really make sure we understand how to be a Super Digital Citizen!

Directions:

- Have each student grab a small selection of papercraft sheets and encourage them to blend the pieces to make their very own super hero.
- Allow plenty of time for students to cut, glue, and color.
- Give students a 5 minute warning to wrap up.
- Separate students into groups of 2-4 and tell them to use their super heroes and leftover supplies to stage a scene in which one superhero sees an act of poor digital citizenship. Then have the superhero fix the problem … and save the day!
- Go around the room, having each student explain their scene to the class.

Lesson Tip

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

Wrap Up (15 min)

Flash Chat: What did we learn?

- What is a good way to act responsibly online?
- What kinds of personal information could you share about yourself without showing your identity?
- What kinds of superpowers or qualities did your digital superheroes have in common?
- What does Spider-Man’s motto “With great power comes great responsibility” mean to you, as someone who uses the internet?

Lesson Tip

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

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a Digital Citizen?
- What do you need to do to be a Digital Citizen?

Assessment (5 min)

Digital Citizenship - Assessment

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

Extended Learning

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

Common Sense Education

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

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **NI** - Networks & the Internet

Lesson 11: Digital Citizenship: Screen Out the Mean

Common Sense Education | Cyberbullying | Unplugged

Overview

This lesson helps children to recognize that it is essential to tell a trusted adult if something online makes them feel angry, sad, or scared.

Students learn that other people can sometimes act like bullies when they are online. They will explore what cyberbullying means and what they can do when they encounter it. After reading a scenario about mean online behavior, students discuss what cyberbullying is, how it can make people feel, and how to respond. Finally, they use their knowledge to create a simple tip sheet on cyberbullying in their journal.

Purpose

Students may not ever have the misfortune of experiencing cyberbullying, but we want to make sure that the students are prepared for and knowledgeable about it, in case they ever witness it during an online situation. Students will learn how to identify cyberbullying and what steps they should take to make it stop. This may become helpful in later puzzles when students have the opportunity to share their work. If someone negatively responds to a student's work, this lesson will provide them with the tools that they need to handle the situation.

Agenda

Warm Up (5 min)

Introduction

Main Activity (35 min)

What Is Cyberbullying?
What to Do About Cyberbullying

Wrap Up (15 min)

Flash Chat: What did we learn?
Journaling

Assessment (5 - 10 min)

Screen Out the Mean - Teacher Prep Guide

Objectives

Students will be able to:

- Analyze online behaviors that could be considered cyberbullying.
- Explain how to deal with a cyberbullying situation.
- Recognize the importance of engaging a trusted adult if the student experienced cyberbullying.

Preparation

- [ ] Review **Screen Out the Mean - Teacher Prep Guide** from Common Sense Education's website.
- [ ] Print out a worksheet from the link above (page 6) for each student.
- [ ] Print out an assessment from the link at the top (page 7) for each student.
- [ ] Make sure every student has a**Think Spot Journal - Reflection Journal**.
- [ ] Print or display the **Online Safety Poster - Student Handout** for the class to see.

Links

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

For the Teacher

- **Screen Out the Mean** - Teacher Prep Guide
- **Common Sense Education** - Website
- **Think Spot Journal** - Reflection Journal

For the Students

- **Online Safety Poster** - Student Handout

Vocabulary

- **Cyberbullying** - Doing something on the internet, usually again and again, to make another personal feel angry, sad, or scared.
- **Online** - Connected to the internet.

Teaching Guide

Warm Up (5 min)

Introduction

Encourage:

Encourage students to share what they know about bullying.

Ask:

- What kinds of things count as bullying?
  - Students should understand that bullying is behavior that is purposefully mean or scary to someone else. For example, making fun of how someone looks, telling lies about them, or threatening to do something bad to them.
- How does bullying make other people feel?
  - Hurt, angry, upset, scared
- What is the best thing to do when you feel bullied, or when you see someone else being bullied?
  - Students should know to always tell a trusted adult when they experience or witness bullying.

Explain:

Students will be learning about a kind of bullying that can take place when they use the internet.

Main Activity (35 min)

What Is Cyberbullying?

Define:

- **Online**: Connected to the internet
- **Cyberbullying**: Doing something on the internet, usually again and again, to make another person feel angry, sad, or scared

Discuss:

Some kids do not go online very much at all, either because of their family's rules or because they do not like it very much. Other kids do go online to do different things.

Ask:

- What do you do online, or what do you think you might like to do?
  - Students may mention activities like sending messages to friends and playing games.

Share:

Most of the time when students go online it is to do fun or interesting things. But sometimes people can be mean to each other online and this is called cyberbullying.

Ask:

- Did you ever see someone make someone else feel bad online?
  - Answers will vary. Remind students to tell what happened, but not to use real names.

Explain:

Tell students that they will be learning more about how cyberbullying occurs, and what to do when it happens to them or to someone they know.

What to Do About Cyberbullying

Discuss:

Read aloud these two scenarios and discuss them briefly with the class.

- Kyle keeps getting instant messages from someone saying means things about him. The person who is sending the messages doesn't use a real name, but Kyle can tell the messages are coming from someone who also makes fun of him at school in gym class.
- Sasha is a new girl at school, and she's making a lot of friends. Then Sasha finds out that another girl sent around an email that had a picture of a cow with Sasha's name on it.

Next, pass out the **Screen Out the Mean - Teacher Prep Guide** worksheet from page 6. Read aloud the story at the top and ask students to work in pairs or groups to finish the worksheet.

Discussion

Questions to stimulate discussion include:

- What do you think happened to Jada's game?
- How do you think Jada, Kyle, or Sasha felt when these things happened to them?
- How do you know if someone is cyberbullying you?
- Why do you think it is important to stop using the computer when someone starts cyberbullying you?
  - It's possible that if students stay online, the cyberbullying may continue or get worse.

Teacher Tip:

These scenarios can be read all at once and discussed as a whole, or be read and discussed individually.

Ask the class to discuss to Jada's story. Tell the class there are specific steps to handling a cyberbully.

- Jada should STOP using the computer.
- Jada should TELL an adult she trusts what happened.
- Jada should not go back online or return to the pony website until an adult says it is OK.
- If Jada and Michael are good friends, Jada may want to tell Michael how his actions made her feel after she gets help from an adult.
- If Michael continues cyberbullying her, she should play with other kids who don't cyberbully others.

In general, there are four steps students should take if they or someone they know are experiencing cyberbullying.

1. Stop using the computer until it is safe.
2. Tell an adult you trust.
3. Go online only when a trusted adult says it is okay.
4. Play online only with kids who you know and are nice.

Wrap Up (15 min)

Flash Chat: What did we learn?

Ask:

- What is cyberbullying. How does it make people feel?
  - Students should recognize that cyberbullying is any kind of online behavior that makes people feel sad, scared, angry or upset.
- What four things can you do to help stop cyberbullying?

  S. **Stop** using the computer until it is safe.

  T. **Tell** an adult you trust.

  O. Go **Online** only when a trusted adult says it is okay.

  P. **Play** online only with kids who are nice.
- What is the most important thing to do if someone starts cyberbullying you?
  - Telling a trusted adult is the most important response whenever someone makes them feel sad, scared, or angry online.

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Write down the names of some trusted adults you can go to if you ever feel bullied.
- What are the four steps you should take if you or someone you know is being cyberbullied.

Assessment (5 - 10 min)

Screen Out the Mean - Teacher Prep Guide

Pass out an assessment to each student. Allow students a few minutes to complete it then review the answers (page 9 of the link above) with the class. If there's time, allow for a discussion about the questions.

Lesson 12: Events: The Big Event

Unplugged | Events

Overview

Students will soon learn that events are a great way to add flexibility to a pre-written algorithm. Sometimes you want your program to be able to respond to the user exactly when the user wants it to. Events can make your program more interesting and interactive.

Purpose

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

Agenda

Warm Up (15 min)

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

Main Activity (15 min)

- **The Big Event - Worksheet**

Wrap Up (10 min)

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

Assessment (10 min)

- **The Big Event - Assessment**

Extended Learning

Objectives

Students will be able to:

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

Preparation

- [ ] Watch the **The Big Event - Teacher Video**.
- [ ] Print one **The Big Event - Worksheet** and Event Controller.
- [ ] Print one **The Big Event - 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

- **The Big Event** - Unplugged Video (**download**)
- **The Big Event** - Teacher Video
- **The Big Event** - Worksheet
- **The Big Event** - Assessment
- **The Big Event** - Assessment Answer Key

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has one new and important vocabulary word:

Event - Say it with me: E-vent

An event is an action that causes something to happen.

A Series of Events

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

Main Activity (15 min)

The Big Event - Worksheet

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

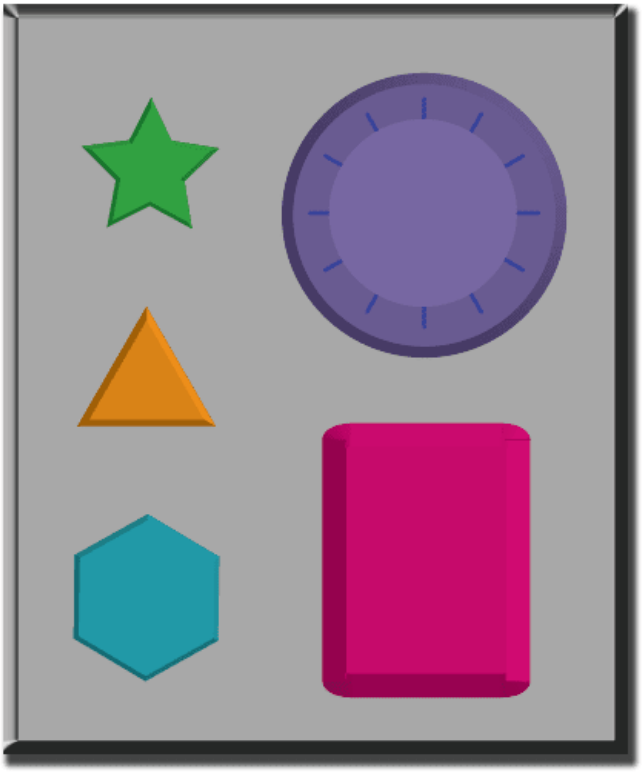
Lesson Tip

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

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

Directions:

- Project the Event Controller onto your classroom screen.



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

Wrap Up (10 min)

Flash Chat: What did we learn?

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

Assessment (10 min)

The Big Event - Assessment

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

Extended Learning

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

One Person's Event is Another One's Reaction

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

Eventopalooza

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

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 13: Events in Star Wars

Star Wars | Event

Overview

In this lesson, students will practice using events to build a game that they can share online. Featuring R2-D2 and other Star Wars characters, students will be guided through events, then given space to create their own game.

Purpose

CS Fundamentals is not simply about teaching computer science, it is about making computer science fun and exciting. In this series, students will learn about events using popular characters from Star Wars. These puzzles blur the lines between "learning" and "fun". Also, students will learn to recognize regular programming practices in games so that when they play games at home, they can see common computer science principles being used.

Agenda

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

Bridging Activity - Events (15 min)
- **Unplugged Activity Using Paper Blocks**
- **Preview of Online Puzzles as a Class**

Main Activity (30 min)
- **CSF Express Course - Website**

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

Objectives

Students will be able to:

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

Preparation

- [ ] Play through puzzles in **CSF Express Course - Website** and find any potential problem areas.
- [ ] 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 E Online Puzzles** - Website
- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **Think Spot Journal** - Reflection Journal

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Introduction

In a class discussion, ask the students what their favorite video game is (you might need to remind the students to only use games that are classroom appropriate). Ask the students what their favorite part of the game is.

Most of the time, students will respond with some kind of event. When you recognize a student response that describes an event, ask the student to describe it further.

Once the student is done describing their fun, take a minute to relate it back to the definition of an event.

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

Ask the students to try and relate some of their favorite parts of video games and how they can be described as events. Have them pair share and discuss the differences between their events and their partner's.

Teacher Tip

If you're not quite sure if a student's response describes an event, try to break down the response. Is there an action and a response?

For example:

- Crossing the finish line and having on screen characters congratulate you
- Finding a big pot of treasure (or other item) and watching your inventory grow
- Buying new items from the game's store and having the item to use
- Pressing the buttons on a game controller and having your character do something cool

Bridging Activity - Events (15 min)

This activity will help bring the unplugged concepts from "The Big Event" 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

Using the remote from the **The Big Event - Worksheet** and **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives**, gather your class to reprise the activity from the previous lesson. Ask the class "when the teal button is pushed, what do we do?" then fill in one of the when event blocks and one of the blue action blocks accordingly. Make sure that the students understand that the when blocks need to be on top of the blue block and they need to touch in order for the program to run. Play out the rest of the lesson from "The Big Event", referencing the blocks along the way.

Preview of Online Puzzles as a Class

Pull a lesson from the corresponding online stage on **CSF Express Course - Website** , we recommend a puzzle early on in the progression, such as the second puzzle. Ask the students what should happen when the up arrow key on the keyboard is pressed. Explain that the character in this game should move in the direction of the arrow on the keyboard.

Complete the puzzle with the class and allow time for a quick discussion on what was and wasn't an event. For every event, ask the students what the action corresponding to this event is.

Main Activity (30 min)

CSF Express Course - Website

Students will likely be very excited to make their own Star Wars game at the end of this set of puzzles. If there's time, ask them to plan out what they want the game to do. The planning and preparation will help the students better recognize the key concepts this lesson is trying to teach. Encourage the students to share and remix each other's games at the end of this lesson.

Teacher Tip

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?
- Give an example of an event you used in your program today?
- Why is it important not to share private information online? How do you know if information is private?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 14: Events with Flappy

Event | Flappy

Overview

In this special stage, students get to build their own Flappy Bird game by using event handlers to detect mouse clicks and object collisions. At the end of the level, students will be able to customize their game by changing the visuals or rules.

Purpose

Events are very common in computer programs. In this lesson, students will further develop their understanding of events by making a Flappy Bird game. Students will learn to make their character move across the screen, make noises, and react to obstacles based on user-initiated events.

Agenda

Warm Up (10 min)
- **Introduction**

Bridging Activity - Events (10 min)
- **Unplugged Activity Using Paper Blocks**
- **Preview of Online Puzzles as a Class**

Main Activity (30 min)
- **CSF Express Course - Website**

Wrap Up (10 - 15 min)
- **Journaling**

Extended Learning

Objectives

Students will be able to:

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

Preparation

- [ ] Play through **Course C Online Puzzles - Website** in stage 11 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

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **Think Spot Journal** - Reflection Journal

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

- Review "The Big Event" activity with students:
  - What did we "program" the button click events to do?
- Now we're going to add events to our coding. Specifically, we're going to create an event for clicking the mouse and one for when the bird hits an object like the ground or an obstacle. When have you seen a character touch another object as an event in games?

Lesson Tip

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

Bridging Activity - Events (10 min)

This activity will help bring the unplugged concepts from "The Big Event" 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

Using the remote from the **The Big Event - Worksheet** and **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives**, gather your class to reprise the activity from the previous lesson. Ask the class "when the teal button is pushed, what do we do?" then fill in one of the `when` event blocks and one of the blue action blocks accordingly. Make sure that the students understand that the `when` blocks need to be on top of the blue block and they need to touch in order for the program to run.

Preview of Online Puzzles as a Class

Pull a lesson from the corresponding online stage, we recommend puzzle 2. Ask the students what should happen when the Flappy Bird runs into something like the ground or an obstacle. Explain that Flappy in this game will move forward with a click of the mouse and the game will end if Flappy runs into anything.

Complete the puzzle with the class and allow time for a quick discussion on what was and wasn't an event. For every event, ask the students what the action corresponding to this event is.

Main Activity (30 min)

CSF Express Course - Website

In the final stage of this lesson students are able to tweak their game to make it unique - encourage them to see how different they can make each game within the constraints provided. If the class doesn't use **Pair Programming - Student Video**, then tell students to go around and look at other student's games. Otherwise, have students discuss and try out different ways to set up their game with their partner.

Teacher Tip

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 (10 - 15 min)

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 about today's lesson?
- What did you do to make your game unique?
- Draw out a game you want to make in the future.

Extended Learning

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

Look Under the Hood

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

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

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 15: Events in Bounce

Event | Bounce

Overview

In this online activity, students will learn what events are, and how computers use them in programs like video games. Students will work through puzzles making the program react to events (like arrow buttons being pressed.) At the end of the puzzle, students will have the opportunity to customize their game with different speeds and sounds.

Purpose

Events are very common in computer programs, especially in video games.

In this lesson, students will develop their understanding of events by making a sports-based game. Students will learn to make their paddle move according to arrow keys, and make noises when objects collide. At the very end, they will get to customize their game to make it more unique!

Agenda

Warm Up (10 min)
- **Introduction**

Main Activity (30 min)
- **CSF Express Course - Website**

Wrap Up (10 min)
- **Journaling**

Extended Learning

Objectives

Students will be able to:

- Identify actions that correlate to input events.
- Create an interactive game using sequence and event-handlers.
- Share a creative artifact with other students.

Preparation

- [ ] Make sure every student has a**Think Spot Journal - Reflection Journal**.
- [ ] Play through **Course D Online Puzzles - Website** in stage 3 to find any potential problem areas for your class.
- [ ] Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.

Links

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

For the Teacher

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **Think Spot Journal** - Reflection Journal

Vocabulary

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

Teaching Guide

Warm Up (10 min)

Introduction

Ask the students to come sit down near you. Now tell them to all stand up!

Tell the students what you just did was declare an event and an action. When you say to sit down, it is an event. The action responding to this event is the class sitting down. This is the same when you ask the class to stand up. Events and actions are easily identifiable in our lives.

Some other events and actions include:

- Feeling hungry and eating food
- Stubbing your toe and yelling "Ouch!"
- Getting the basketball in the basket and scoring a point for your team!

Ask the class to come up with a couple of more events. Tell them that they will be making a game where the program will have actions associated to events that they code!

Main Activity (30 min)

CSF Express Course - Website

At the end of the set of puzzles, students will have the opportunity to make their game unique. Have the students try new ways to make the game more challenging. For example, try playing with many balls at once, or each time the ball bounces off a wall, launch more balls.

Teacher Tip

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** .

This is what **Justin Trudeau, the Prime Minister of Canada, did when he completed an Hour of Code in 2016**.

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?
- What did you do to make your game super cool?
- What kind of game do you want to code in the future?

Extended Learning

Take Me Out to the Ball Game

Take the students outside to play some sort of ball game. Keep track of events and actions. For example, not dribbling

in basketball results in a traveling foul and the other team gets the ball. In soccer, kicking the ball out of bounds results in the other team kicking the ball in. Getting the ball to the goal results in a point! Make up more events if your students are into it. Have the all of the students yell "Yippee" when the captain of one team scores a point. Have everyone fall to the ground and roll around if a student makes two goals in a row!

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **AP** - Algorithms & Programming

EXPRESS Course

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 29 30 31 32 33 34 35

C O D E

Lesson 16: Conditionals: Conditionals with Cards

Conditionals | Unplugged

Overview

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

Purpose

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

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

Agenda

Warm Up (20 min)

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

Main Activity (20 min)

- **Conditionals with Cards Sample Program - Teacher Prep Guide**

Wrap Up (15 min)

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

Assessment (5 min)

- **Conditionals with Cards - Assessment**

Extended Learning

Objectives

Students will be able to:

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

Preparation

- [ ] Watch the **Conditionals with Cards - Teacher Video**.
- [ ] Watch the **Conditionals with Cards - Lesson in Action Video**.
- [ ] Gather decks of cards or something similar.
- [ ] One **Conditionals with Cards Sample Program - Teacher Prep Guide** for the class to look at.
- [ ] Print one **Conditionals with Cards - Assessment** for each student.
- [ ] Make sure every student has a**Think Spot Journal - 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
- **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:

```
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 17: While Loops in Farmer

While Loops | Loops | Farmer

Overview

By the time students reach this lesson, they should already have plenty of practice using `repeat` loops, so now it's time to mix things up.

While loops are loops that continue to repeat commands while a condition is met. While loops are used when the programmer doesn't know the exact number of times commands need to be repeated, but does know what condition needs to be true in order for the loop to continue repeating. For example, students will be working to fill holes and dig dirt in Farmer. They will not know the size of the holes or the height of the mountains of dirt, but the students will know they need to keep filling the holes and digging the dirt as long as the ground is not flat.

Purpose

As your students continue to deepen their knowledge of loops, they will come across problems where a command needs to be repeated, but it is unknown how many times it needs to be repeated. This is where **while loops** come in. In today's lesson, students will develop a beginner's understanding of condition-based loops and also expand their knowledge of loops in general.

Agenda

Warm Up (10 min)
- **Introduction**

Bridging Activity - Conditionals (15 min)
- **Unplugged Activity Using Paper Blocks**
- **Preview of Online Puzzles**

Main Activity (30 min)
- **CSF Express Course - Website**

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

Extended Learning

Objectives

Students will be able to:

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

Preparation

- [ ] Play through **Course D 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

- **CSF Express Course** - Website
- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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.
- **Loop** - The action of doing something over

and over again.

- **Repeat** - Do something again
- **While Loop** - A loop that continues to repeat while a condition is true.

Teaching Guide

Warm Up (10 min)

Introduction

Use "while" in a sentence in front of the students. Ask the students what the word "while" means. If you were to say "while there is a hole, fill it with dirt" what would they do? How long would they do that?

When you use a word like “while”, you are relying on a condition to tell the computer how long the loop should run. A condition is a statement that is tested and found to be true or false. In the case above, the condition is if there is a hole. It’s only possible for there to be a hole or for there not to be a hole, thus the statement is only ever true or false.

Tell the students they will be learning about a new kind of loop. Previously, students only used loops to repeat a command a certain number of times. Here, they won't always know how many times to repeat the command, however, they will know when to stop or when to keep going. While loops allow the programmer to repeat a command as long as a condition is still true. In the previous example, the condition is the existence of a hole.

If there's time, have the students discuss other times using a while loop would be useful. Examples include:

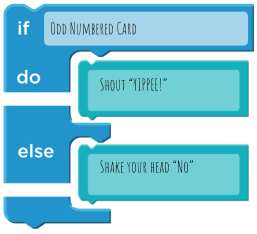
- Running toward a ball **while** it is in front of you.
- Filling a glass **while** it has space for more liquid.
- Walk forward **while** there is a path ahead.

Bridging Activity - Conditionals (15 min)

This activity will help bring the unplugged concepts from "Conditionals With Cards" 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

Print and cut out 2-3 `if / else` and blank action blocks from **Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives** and pull out a deck of cards. Ask the class to come up with a couple of conditionals to use with the deck of cards like they did in "Conditionals with Cards." When the conditionals have been decided on as a class, fill in the blank part of the `if` block with the various card values that the kids came up with. Examples include "King of Hearts", "Even Numbered", or "Diamonds". Fill in the action blocks with the actions the students came up with. Make sure the students know the action blocks need to be directly under the `if` or `else` block. Below is an example.



Now shuffle the deck of cards and play "Conditionals with Cards" again. Flip through the deck card-by-card, reacting to cards if a conditional has been made for it.

Preview of Online Puzzles

Pull up a puzzle from Course D, we recommend puzzle 9.

- Ask the class what the bee should do when it gets to the cloud.
  - The bee should use a conditional to check for a flower or a honeycomb.
- Use the `if at flower / else` block. Ask the class what the bee should do if there's a flower. If there's not a flower, there

will be a honeycomb. What should the bee do then?
    - The bee should get nectar if there is a flower and make honey if there is a honeycomb.

Fill in the rest of the code and press Run . Discuss with the class why this worked.

Main Activity (30 min)

CSF Express Course - Website

While loops are not always a difficult concept for students to understand, but if you think your class might struggle with these puzzles, we recommend **Pair Programming - Student Video** . This will allow students to bounce ideas of each other before implementing the code. Pair programming works to increase confidence and understanding with topics like while loops.

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 the difference between a while loop and a normal repeat loop?
- Give an example of a puzzle where you would use a while loop, but not use a repeat loop. Can you give an example of a puzzle where you would use a repeat loop, but not a while loop?

Extended Learning

While Simon Says

Go out to a large playing field and have the students stand in a line facing you. Make sure every student can see you. Declare a couple of "while loops" such as:

- While my right hand is up, you can walk toward me
- While I cover my eyes, you can skip toward me
- While my head is turned to the right, you have to walk backwards

The first student to get to you wins. If there's time, let other students be "Simon" at the front of the class.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 18: Conditionals & Loops in Maze

Conditional | Loop | Maze | Angry Bird | Zombie

Overview

In this lesson, students will be pairing together two key concepts: loops and conditionals. This set of puzzles bridges the gaps in understanding that occur when working on puzzles that use multiple kinds of blocks. By bringing two ideas together, students will create more complex code that shows both impressive creativity and critical thinking!

Purpose

This set of puzzles will work to solidify and build on the knowledge of conditionals and loops. By pairing these concepts together, students will be able to explore the potential for creating complex and innovative programs.

Agenda

Warm Up (10 min)
- **Introduction**

Main Activity (30 min)
- **CSF Express Course - Website**

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

Extended Learning

Objectives

Students will be able to:

- Build programs with the understanding of multiple strategies to implement conditionals.
- Translate spoken language conditional statements and loops into a program.

Preparation

- [ ] Play through **Course D 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

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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.
- **Loop** - The action of doing something over

and over again.

- **Repeat** - Do something again
- **While Loop** - A loop that continues to repeat while a condition is true.

Teaching Guide

Warm Up (10 min)

Introduction

Students will be bringing together the new `while` loops and new `if / else` statements, so this introduction will go over what these blocks do in a general sense.

`While` Loops

Ask the class if they remember what "while" means in coding. Go over the definition of a while loop.

- **While Loop**: A loop that continues to repeat while a condition is true.

In addition to `while` loops, students will practice with `until` loops in this set of puzzles. Explain to the students that `while` loops continue to repeat code while a condition is true, whereas `until` loops continue until a condition is true.

For example, with a `while` loop, the zombie will continue to walk down a path **while** there is a path ahead. With an `until` loop, the zombie will continue to walk forward **until** it reaches the flower at the end of the path. This will be shown in more detail inside the puzzles.

`If / Else` Statements

Ask the class if they remember what "if" and "else" mean in coding. Go over the definition of a conditional.

- **Conditional**: Statements that only run under certain conditions or situations.

Students will be using conditionals to test if there are paths to the left or right. Explain that conditionals are extremely flexible and can be used in a way that your program can adapt to almost any situation.

All Together Now

Ask the class to explain why `while` loops are conditionals. How are they different from `if / else` statements? Open up a discussion on when to use `while` loops and when to use `if / else` statements.

Main Activity (30 min)

CSF Express Course - Website

Bringing together concepts is not easy, but this set of lessons is meant to bridge `if / else` statements and `while` loops together for students to see the endless possibilities of coding when using the pair. If students struggle at all with understanding the similarities or differences between `if / else` statements, `while` loops, or `until` loops, have them go back and practice on previous puzzles that only uses one of the three.

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's the difference between an `until` loop and a `while` loop?
- What do you like using the most: `if / else` statements, `until` loops, or `while` loops? Why?

Extended Learning

Until Simon Says

Go out to a large playing field and have the students stand in a line facing you. Make sure every student can see you. Declare a couple of "until loops" such as:

- Until my right hand is raised up, you can walk toward me
- Until I say "eggplant", you can walk backwards in my direction
- Until I turn my head is to the right, you have to walk like a crab

The first student to get to you wins. If there's time, let other students be "Simon" at the front of the class.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 19: Conditionals in Minecraft

Conditional | Minecraft

Overview

This lesson gives students a chance to learn and practice conditionals. It features characters and settings from Minecraft, and students will complete tasks such as mining and building structures using their programs.

Purpose

This set of puzzles will work to solidify and build on the knowledge of conditionals and loops. By pairing these two concepts together, students will be able to explore the potential for creating complex and innovative programs in a new and exciting environment.

Agenda

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

Main Activity (30 min)
- **CSF Express Course - Website**

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

Extended Learning

Objectives

Students will be able to:

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

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

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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.

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)

CSF Express Course - Website

Students are in for a real treat with this lesson. It's likely most of your students have heard of Minecraft, but give a brief introduction for those that may not know.

Minecraft is a game of cubes. You can play as Alex or Steve as you work through mazes. You'll need to avoid lava, pick up items, and explore in a world made up of cubes of things.

Ask the students if they have ever played Minecraft. If none have, move on to the main activity. If some have, ask those experts to explain the game to the class. If everyone in the class has already played, go ahead and move on to the online 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.

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- What did you enjoy about the puzzles today?
- When did you use conditionals in this lesson? Why did you use them?

Extended Learning

More Minecraft

If you find that your class really enjoys the Minecraft environment,**here are some links to other Minecraft games they can play online**. These games will also teach basic coding skills.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▶ **AP** - Algorithms & Programming

EXPRESS Course 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 29 30 31 32 33 34 35 C O D E

Lesson 20: Conditionals & Loops in Farmer

Conditional | Loop | Farmer

Overview

Students will practice while loops, until loops, and if / else statements. All of these blocks use conditionals. By practicing all three, students will learn to write complex and flexible code.

Purpose

Practicing the use of conditionals in different scenarios helps to develop a student's understanding of what conditionals can do. In the previous lesson, students only used conditionals to move around a maze. In this lesson, students will use conditionals to help the farmer know when to harvest crops. New patterns will emerge and students will use creativity and logical thinking to determine the conditions where code should be run and repeated.

Agenda

Warm Up (5 min)
- **Introduction**

Main Activity (30 min)
- **CSF Express Course - Website**

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

Objectives

Students will be able to:

- Nest conditionals to analyze multiple value conditions using if, else if, else logic.
- Pair a loop and conditional statement together.

Preparation

- [ ] Play through **Course D 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

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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.

- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again
- **While Loop** - A loop that continues to repeat while a condition is true.

Teaching Guide

Warm Up (5 min)

Introduction

Students shouldn't need as much of an introduction to concepts today because they have had practice with them in the previous lesson. Instead, you can share the story of the farmer.

The farmer is trying to harvest crops like pumpkins, lettuce, and corn. However, the farmer has forgotten where she planted these crops, so she needs to check each plant before harvesting.

Main Activity (30 min)

CSF Express Course - Website

Students will continue to work with `if / else` statements, `while` loops, and `until` loops. These puzzles are a bit more challenging, though, so encourage students to stick with them until they can describe what needs to happen for each program.

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 can you see conditionals being useful in programs?
- What if people only spoke in if/else statements? What would be some advantages and disadvantages of this?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 21: Variables: Envelope Variables

Unplugged | Variable

Overview

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

Purpose

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

Agenda

Warm Up (10 min)

- **Vocabulary Introduction**

Main Activity (20 min)

- **Envelope Variables - Worksheet**

Wrap Up (10 min)

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

Assessment (10 min)

- **Envelope Variables - Assessment**

Extended Learning

Objectives

Students will be able to:

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

Preparation

- [ ] Watch the **Envelope Variables - Teacher Video**.
- [ ] Obtain 6 or more blank envelopes for warm up plus some for the main activity.
- [ ] Print one **Envelope Variables - Worksheet** per student.
- [ ] Print one **Envelope Variables - Assessment** for each student.
- [ ] Provide students with envelopes, paper, pens & pencils.
- [ ] Make sure every student has a**Think Spot Journal - 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

- **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 22: Variables in 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 are unknown at the time that you run your program or for values that can change during the execution of a program. These constructs are vital to creating dynamic code because they allow your program to change and grow based on any number of potential modifications. This stage teaches students what variables are, using the most basic capabilities of setting and using them.

Agenda

Warm Up (15 min)

- **Introduction**

Bridging Activity - Variables (15 min)

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

Main Activity (30 min)

- **CSF Express Course - Website**

Wrap Up (15 min)

- **Journaling**

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

- **CSF Express Course** - Website
- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **Think Spot Journal** - Reflection Journal

For the Students

- **Variables** - Student Video (**download**)

Vocabulary

- **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.

- 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?

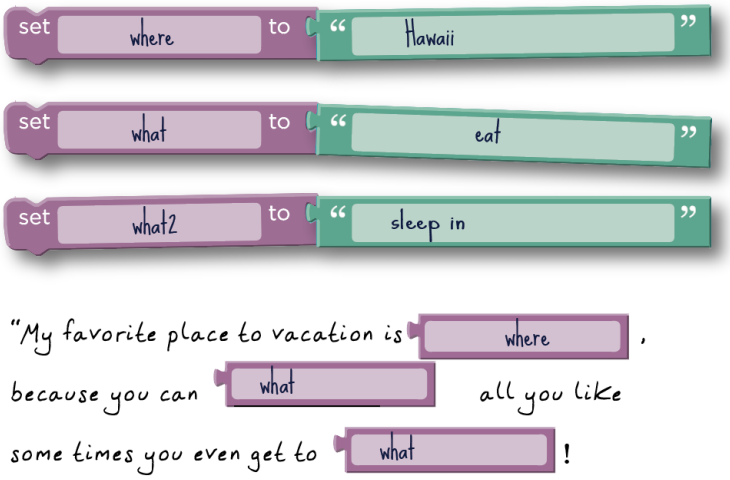
If the class seems interested, continue the discussion. Otherwise, move on to one of the bridging activities.

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. Choose **one** of the following to do with your class:

Unplugged Activity Using Paper Blocks

Start with a sentence or paragraph on the board that contains at least one (and up to five) named blanks. Give paper blocks to each student group. Challenge each group to "set" their variables for each blank. Call on different student groups to share their assignment of each variable and see what happens!



Now, change the sentence to a math equation. What happens to the sentence "X + Y =" when students assign different values to the variables X & Y?

Preview of Online Puzzles as a Class

Display a puzzle to the class. We recommend the 3rd puzzle. Build the code the long way first (use exact numbers for each value, instead of utilizing variables) then suggest that you should try making the squares only 50 pixels. What a pain! What have students learned that will let them give something a name and use it as often as they want later in the program? Go back and add a variable to the beginning. Set the variable to 80, and substitute all occurrences of 80 in the program. Next, change it to 50. That was easy!

Main Activity (30 min)

CSF Express Course - 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.) Once that idea has been presented, it flips to show how you can include a variable for information that changes after the program is run.

Watch out for puzzle #5. 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 (#4) 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 #5.

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

Lesson 23: Variables in Play Lab

Variable | Play Lab

Overview

Students will get further practice with variables in this lesson by creating scenes in Play Lab. Students will work with user input to set the values of their variables, then get space to create their own mini-project with variables.

Purpose

This lesson lets students use variables to display phrases or conversations based on user input. This lesson serves as a wonderful practice exercise for variables in programming, with an extra dose of creativity! At the end of the puzzle sequence, students will be presented with the opportunity to share their projects with family and friends.

Agenda

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

Main Activity (30 min)
- **CSF Express Course - Website**

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

Objectives

Students will be able to:

- Use variables to hold words and phrases.
- Use variables in conjunction with character prompts.

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

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **Think Spot Journal** - Reflection Journal

Vocabulary

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

Teaching Guide

Warm Up (15 min)

Introduction

Ask the class to raise their hand and say what their favorite food is. Choose a couple of students to respond. On a display (either a whiteboard or a poster) write:

(name of the student) likes (favorite food)

Example: Kiki likes pasta.

After a couple of students, ask the class if you could use variables to continue this for the rest of the class. Ask them where the variables might go and what they might be called. Once the discussion is done write:

[name] likes [food]

on the same display.

Ask the students what is a variable and what isn't. How do they know? What else could they use variables for in sentences like these? (examples: favorite color, hometown, number of siblings, etc)

Main Activity (30 min)

CSF Express Course - Website

Your students have already been introduced to variables, but if you find one struggling with the idea, remind them to ask their peers before coming to the teacher for help. This stimulates discussion and encourages a community of learning.

Teacher Tip

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 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 24: For Loops: 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

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** - Teacher Video
- **For Loop Fun** - Lesson in Action Video
- **For Loop Fun** - Worksheet
- **For Loop Fun** - Worksheet Answer Key

Vocabulary

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

Teaching Guide

Warm Up (20 min)

Vocabulary

This lesson has one new and important word:

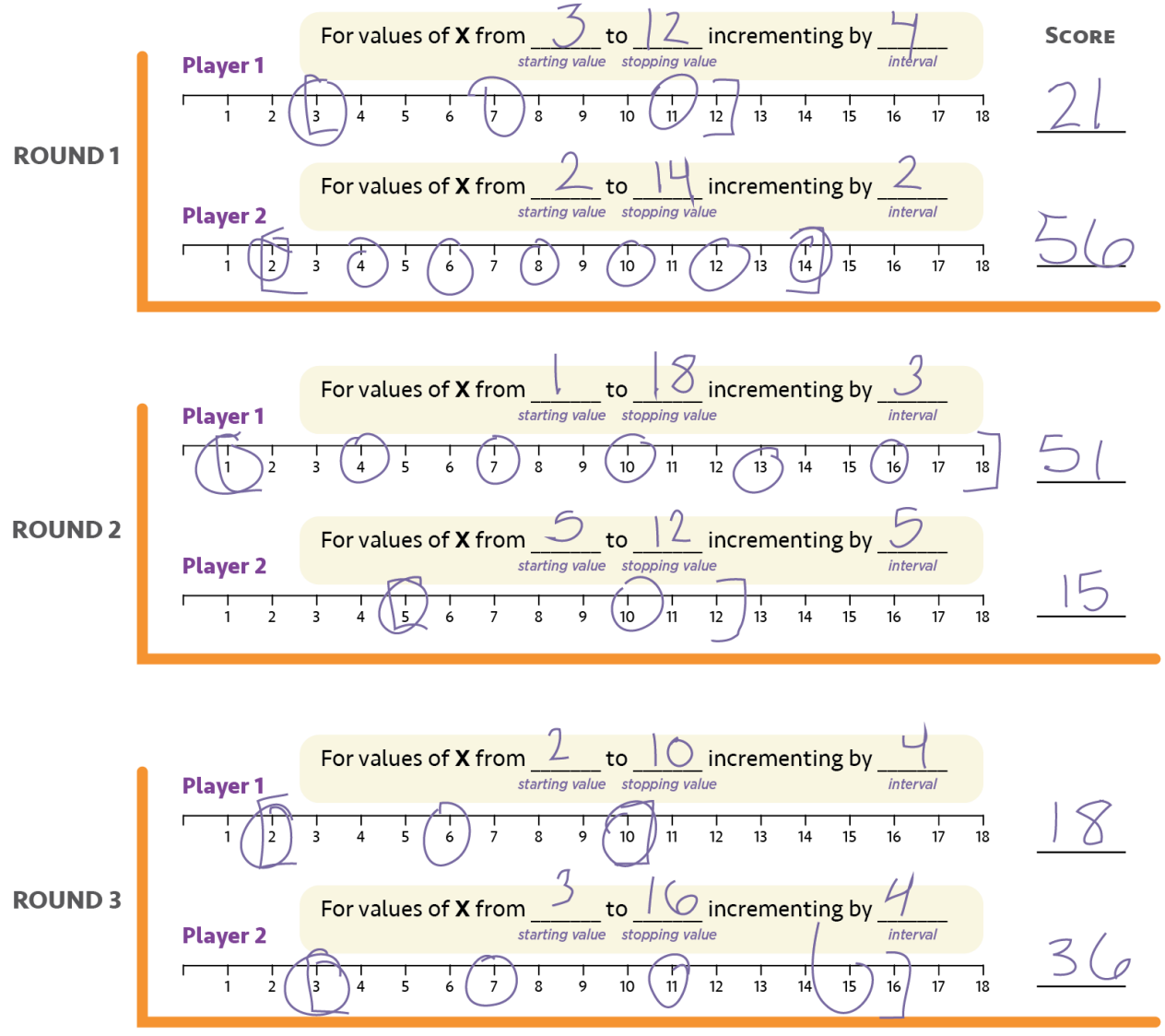
- **For Loop** - Say it with me: For-Loop

Loops that have a predetermined beginning, end, and 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

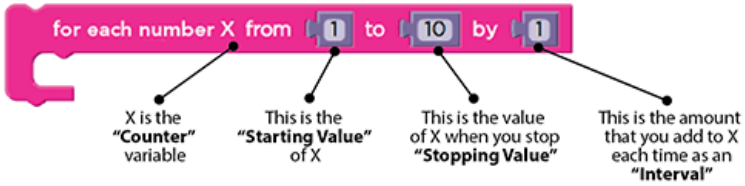
Here, you can jump right into a sample of the game (example in English)



Main Activity (20 min)

For Loop Fun - Worksheet

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



for Loop block (in English)

Directions:

- Divide students into pairs
- To start the round, each student rolls three times:
  - One die to determine the starting value of X
  - Three dice to determine the stopping value for X
  - One die to determine the 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

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;
}
```

It may be difficult for young students to understand this written in pseudocode, but it may be helpful to have you explain out loud (and perhaps with a diagram) what they will be using as the content of a for loop.

Wrap Up (15 min)

Flash Chat: What did we learn?

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

Lesson Tip

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

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a for loop?

- Why would you use a `for` loop instead of a `repeat` loop or a `while` loop?

Assessment (5 min)

For Loop Fun - Assessment

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

Extended Learning

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

Run it Backward

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

Hop Scotch

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

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 25: For Loops in 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)
- **CSF Express Course - Website**

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

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

- **CSF Express Course** - Website
- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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)

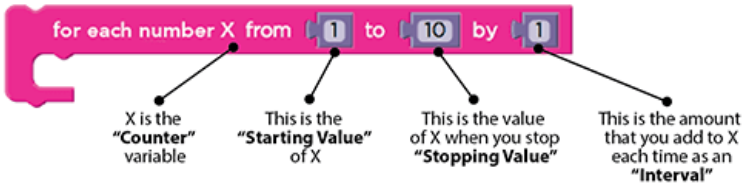
This activity will help bring the unplugged concepts from "For Loop Fun" into the online world that the students are moving into. Choose **one** of the following to do with your class:

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.



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)

CSF Express Course - 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 26: For Loops in 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)
- **CSF Express Course - Website**

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

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

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **Think Spot Journal** - Reflection Journal

Vocabulary

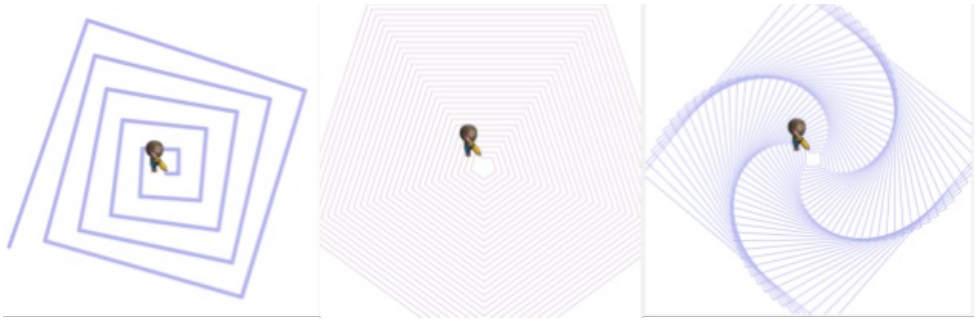
- **For Loop** - Loops that have a predetermined beginning, end, and increment (step interval).

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)

CSF Express Course - 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 27: Functions: Songwriting with Parameters

Unplugged | Function | Parameter

Overview

One of the most magnificent structures in the computer science world is the function. Functions (sometimes called procedures) are mini programs that you can use over and over inside of your bigger program. This lesson will help students intuitively understand why combining chunks of code into functions is such a helpful practice, and how they can use those structures even when chunks of code are slightly different.

Purpose

Using functions helps simplify code and develops the student's ability to organize their program. Parameters will help the students customize their functions so that they can be used for patterns that are similar, though not identical. Students will quickly recognize that writing functions will make their long programs easier to read and easier to debug if something goes wrong.

Agenda

Warm Up (15 min)

- **Vocabulary**
- **Sing a Song**

Main Activity (20 min)

- **Functions Unplugged: Songwriting With Parameters - Worksheet**

Wrap Up (15 min)

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

Assessment (5 min)

- **Functions Unplugged: Songwriting With Parameters - Assessment**

Extended Learning

Objectives

Students will be able to:

- Modify functions to accept parameters.
- Describe how functions and parameters can make programs easier to write.

Preparation

- [ ] Watch the **Songwriting and Songwriting with Parameters (Functions) - Teacher Video**.
- [ ] Watch the **Functions Unplugged: Songwriting With Parameters - Lesson in Action Video**.
- [ ] Print several **Functions Unplugged: Songwriting With Parameters - Worksheet** for each group.
- [ ] Print one **Functions Unplugged: Songwriting With Parameters - Assessment** for each student.
- [ ] Access to the internet, or pre-downloaded songs and lyrics for activity.
- [ ] Make sure every student has a**Think Spot Journal - Reflection Journal**.

Links

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

For the Teacher

- **Songwriting With Parameters** - Unplugged Video (**download**)
- **Songwriting and Songwriting with Parameters (Functions)** - Teacher Video
- **Functions Unplugged: Songwriting With Parameters** - Lesson in Action

Video

- **Functions Unplugged: Songwriting With Parameters** - Worksheet
- **Functions Unplugged: Songwriting with Parameters** - Worksheet Answer Key
- **Functions Unplugged: Songwriting With Parameters** - Assessment
- **Functions Unplugged: Songwriting with Parameters** - Assessment Answer Key
- **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.
- **Parameter** - An extra piece of information passed to a function to customize it for a specific need

Teaching Guide

Warm Up (15 min)

Vocabulary

This lesson has two new and important words:

- **Function** - Say it with me: Func-shun

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

- **Parameter** - Say it with me: Pa-ram-eh-ter

An extra piece of information that you pass to the function to customize it for a specific need.

Sing a Song

- Let the class know that today is song day!
- We're going to learn a song together.
  - Start with a simple song either written out or projected on the screen
  - Point to the chorus and be sure that the class knows how it goes before you begin on the rest of the song
  - Blast through the song, singing it with them in the beginning, then see what happens when you get to the part where it calls the chorus

Chorus:
Little bunny Foo Foo
Hopping through the forest
Scooping up the field mice
And bopping 'em on the head
Down came the Fairy
And she said
"Little bunny Foo Foo
I don't wanna see you
Scooping up the field mice
And bopping 'em on the head"

Song:
Chorus

I'll give you 3 chances.
Then I'll turn you into a goon!
The next day. . .

Chorus

I'll give you 2 chances.
Then I'll turn you into a goon!
The next day. . .

Chorus

I'll give you 1 chance.
Then I'll turn you into a goon!
The next day. . .

Chorus

"I gave you two chances.
Now I'll turn you into a goon!"

Teaching Tip

Little Bunny Foo Foo is being used here as an example only. If your students know this song, feel free to use it. Otherwise, choose an appropriate song that they might be more familiar with (either from music class or the radio.)

(POOF!)
And the moral of the story is:
Hare today, goon tomorrow!

- It's quite likely that the majority of the class will sing the lyrics for the chorus when you point to that bit.
  - Stop the song once that happens, and explicitly highlight what just happened
    - You defined the chorus
    - You called the chorus
    - They sang the chorus
- Ask the class why they suppose you only wrote the chorus once at the top of the paper instead of writing it over and over in each place where it is supposed to be sung.
  - What are other benefits of only writing the chorus once when you sing it many times?

Now, imagine that this song is a computer program. Defining a title (like "chorus") for a little piece of code that you use over and over again is called creating a function. This is helpful to computer scientists for the some of the same reasons that it is helpful to songwriters.

Lesson Tip

To add more interest, you can look up the lyrics for some popular songs on the Internet. Show the students that the standard for repeating lyrics is to define the chorus at the top and call it from within the body of the song.

- It saves time not having to write all the code over and over in the program
- If you make a mistake, you only have to change it one place
- The program feels less complicated with the repeating pieces defined just once at the top

What about songs where the chorus isn't exactly the same every time? You can still use a chorus, but you have to have a way to let the singer know what special words you will use for each verse.

- These special words are called parameters.
- In programming, parameters are passed as special instructions to functions like this:

```
chorus(parameter1, parameter2)
```

Feel like this is starting to get complicated? Don't worry. We're going to play with songs a little more to try to really understand how this technique is used!

Main Activity (20 min)

Functions Unplugged: Songwriting With Parameters - Worksheet

A fantastic way to compare functions to something we see in our everyday lives is to look at songs. Songs often have certain groups of lyrics that repeat over and over. We call that a chorus.

Directions:

1. Divide into groups of 4, 5, or 6.
2. Give each group several copies of the Songwriting Worksheet
3. Play a short song for the class that contains a clear chorus that does not change from verse to verse.
4. Challenge the class to identify (and write down) the chorus.
5. Compare results from each group. Did everyone get the same thing?
6. Try the activity again, but this time with a song that changes during each repetition of the chorus. Good examples are: Old MacDonald, Baby Bumblebee, or The Hokey Pokey

Discuss with the class:

- Can the students identify a chorus when some words change?
- How might they use the same idea of calling a chorus when the chorus is different from verse to verse?
- These changing words and phrases are called “parameters” and you can pass them into the chorus like this:

chorus(cow, moo)
- Play this game over and over until the class has little trouble identifying the choruses.

It is often easier just to have the class listen to (or watch) the song, then vote on what the chorus is by singing it together, rather than writing the whole thing down. If you choose this method, consider having the class do a written chorus for the final song selection to be sure that the visual learners get proper reinforcement.

Lesson Tip

It's most exciting for students to do this lesson with popular music from the radio, but if you're having a hard time finding appropriate songs where the lyrics repeat exactly, here are a few timeless options:

- **5 Little Monkeys**
- **Old MacDonald**
- **Hokey Pokey**
- **BINGO**
- **Baby Bumble Bee**

Wrap Up (15 min)

Flash Chat: What did we learn?

- Would you rather write lyrics over and over again or define a chorus?
- Do you think it's possible to make multiple choruses for the same song?
- Does it make sense to make a new chorus for every time it's needed in a song?

Lesson Tip

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

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- How do you see functions being helpful in computer science?
- Describe why parameters are helpful when writing the lyrics for a song where the chorus changes slightly.

Assessment (5 min)

Functions Unplugged: Songwriting With Parameters - Assessment

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

Extended Learning

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

Create Your Song

- Start by creating a chorus together, then repeat it between verses of a song that you develop around it.
- Make a change to the chorus, and ponder how much easier it is to change in just one place.
- Change the chorus again, making it much longer than it was originally.
- Add a second chorus and alternate between them in your verses.
- Add parameters to one of your choruses and see how many more options you have.

Songwriting a Program

- What if we acted out songs instead of singing them? All of the sudden, our chorus would be a function of repeated actions, rather than words.
- Use the concepts of the arrows from the Graph Paper Programming lesson and create a program with lots of repeating instructions.
  - Circle those repeating actions so that the class can see where they are.
  - Define a function called "Chorus" above the program.
  - Cross out everywhere the repeating actions appear in the program and write "Chorus" instead.
- Repeat until the class can go through this process fairly undirected.
- Can you figure out how to pass parameters in this exercise?

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 28: Functions in Bee

Function | Bee

Overview

This lesson teaches students how to create simple functions using our sophisticated “modal” function editor, preparing the way for them to incorporate parameters in future lessons.

Purpose

Students will discover the versatility of programming by practicing functions in different environments. Here, students will recognize patterns in the bee's maze. The bee will need to navigate the play area, collect nectar, and make honey. Students will learn to organize their programs and create functions for repeated code.

Agenda

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

Bridging Activity - Functions (15 min)
- **Unplugged Activity Using Paper Blocks**
- **Previewing Online Puzzles as a Class**

Main Activity (30 min)
- **CSF Express Course - Website**

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

Extended Learning

Objectives

Students will be able to:

- Categorize and generalize code into useful functions.
- Recognize when a function could help to simplify 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

- **CSF Express Course** - Website
- **Unplugged Blockly Blocks (Grades 2 - 5)** - Manipulatives (**download**)
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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.
- **Parameter** - An extra piece of information

passed to a function to customize it for a specific need

Teaching Guide

Warm Up (15 min)

Introduction

Students that have done **Course E Online Puzzles - Website** will already have experience using functions to solve online puzzles, but the function editor in this course is slightly different. Let students know that they will get a quick review of simple functions before moving in to more difficult challenges with the new "modal" editor.

For the students who are less familiar with using functions online, start by reviewing the vocabulary words from the "Functions Unplugged: Songwriting with Parameters".

- **Function** - Say it with me: Func-shun

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

- **Parameter** - Say it with me: Pa-ram-eh-ter

An extra piece of information that you pass to the function to customize it for a specific need.

Tell the class that there are two main components to using functions with parameters.

1. **The Declaration**: Function declarations are what create a function. In a function declaration, you fill in the function with code and you give the function a name. Inside the function declaration you should note where the parameter is used inside the function code. You must declare a function before you can use it.
2. **The Call**: Function calls are what makes the program run the code in the function. To call a function, you place the name of the function in your program with a value for the parameter. Make sure your function is properly defined (with a parameter) before calling it in your program.

The class can use songwriting as an example to understand these two components. In the unplugged activity, the function containing the lyrics to the chorus was named "chorus". When we first made this function, we circled the lyrics that would go in the function. Once we named the function, we could read through the lyrics and replace the repeated chorus lyrics with a function call to "chorus".

Continue the conversation until students have a basic understanding of functions being declared and called. If students don't get to this point, make sure to do one of the bridging activities before moving into the Code.org puzzles.

Note: Students will not be using parameters in their functions today. However, it's good to review what parameters are and why they are used for next time.

Bridging Activity - Functions (15 min)

This activity will help bring the unplugged concepts from “Functions Unplugged: Songwriting with Parameters” 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

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

Ask the students to cross out any part of the song that can be made into a function, even if it has a couple of different words (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?

Lesson Tip

Function blocks:

The block to the left is a function declaration, a block that students will name and 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.

Previewing Online Puzzles as a Class

Pull up a puzzle from **Course F Online Puzzles - Website**. We recommend the 12th puzzle for this activity. 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.

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)

CSF Express Course - Website

Students may benefit from writing code without functions then create functions from the repeated code. If students don't enjoy doing this in the Code.org workspace, we recommend providing paper and pencils for students to write (or draw) out their ideas.

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 did functions help the bee collect nectar and make honey more efficiently?
- Can you imagine using parameters in these puzzles? If so, explain how. If not, why not?

Extended Learning

Draw by Functions

Break the class into groups of 2-3 students. Have each group write a function that draws some kind of shape and a program that uses that function. Depending on the creativity or focus the groups, students might need to be assigned a shape to create. Once every group is done, have the groups switch programs. On a separate piece of paper, each group should draw what the program creates. The groups should then return the programs and drawings to the original group.

Did every group get the drawing they expected? If not, what went wrong? Have the class go through the debugging process and try again.

Standards Alignment

CSTA K-12 Computer Science Standards

- ▸ **AP** - Algorithms & Programming

Lesson 29: Functions with Parameters in Artist

Function | Parameter | Artist

Overview

In this lesson, students continue working with functions with and without parameters. Students will get the chance to create their own drawings before modifying functions in a freeplay level.

Purpose

This lesson is providing students with a space to create something that they are proud of.

These puzzles allow students to create complex images by building off of previous, more simple projects. At the end of this lesson, students will have images to be proud of.

Agenda

Warm Up (10 min)
- **Introduction**

Main Activity (30 min)
- **CSF Express Course - Website**

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

Objectives

Students will be able to:

- Practice abstraction with the use of parameters.
- Recognize when a function could help to simplify a program.

Preparation

- [ ] Play through **Course D 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

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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.
- **Parameter** - An extra piece of information passed to a function to customize it for a

specific need

Teaching Guide

Warm Up (10 min)

Introduction

Ask the class if they recall using parameters in "Functions Unplugged: Songwriting with Parameters".

- What does it mean to "use a parameter" with functions?
  - Using a parameter means the function takes in a variable value that can specialize the function. In "Songwriting with Parameters" this means we can change the repeated lyrics to be a little different everytime.
- Why would we use a parameter with a function?
  - We use parameters with functions so that we don't have to write multiple functions that are very similar. If we wanted to draw three squares with three different side lengths, we would only have to write one function with a parameter versus three different functions without parameters.
- Why don't we always use parameters with functions?
  - We don't ALWAYS need a customizable function. Sometimes functions are just a handy way to reuse identical code in multiple places.

Tell the class that they will be making some awesome drawings in Artist using functions with parameters!

Main Activity (30 min)

CSF Express Course - Website

Ask the students to close their eyes and raise a hand. If they are feeling really good about using parameters, have all fingers open (like a high five). If they don't feel very good about using parameters, have them raise a fist. If they are feeling somewhere in between, have them raise one, two, three, or four fingers on their hand.

With that, determine if your class will need more practice with functions before moving on to the online puzzles. If only a small portion of your class isn't feeling great about using parameters, make sure to implement pair programming in this lesson.

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?
- Sketch out a drawing you made today. Can you write the code needed to create this?
- Draw a picture you would like to create with code. Try writing or drafting the code that would make that drawing.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ **AP** - Algorithms & Programming

Lesson 30: Functions with Parameters in Bee

Function | Parameter | Bee

Overview

This lesson features the bee environment, and continues along the concept of functions with parameters from the previous Artist stage. Students will practice writing and using functions to follow complex paths and collect patterns of nectar and honey.

Purpose

Functions are incredibly important in computer science for many reasons. The ability to break down and categorize code will become immensely important as the programs your students write become more and more complex. Functions with parameters require an extra level of skill. Using functions with parameters teaches your students to recognize when a function is needed and if that function can be generalized enough to be used for multiple cases. This lesson, along with the previous lessons on functions with parameters, builds a strong set of critical thinking and problem solving skills.

Agenda

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

Main Activity (30 min)
- **CSF Express Course - Website**

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

Objectives

Students will be able to:

- Recognize repeated tasks that need to be specialized on a case-by-case basis and create functions to efficiently run these tasks.
- Use pre-built functions with parameters to complete commonly repeated tasks.

Preparation

- [ ] Play through the **Course F Online Puzzles - Website** associated with this level 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

- **CSF Express Course** - Website
- **CS Fundamentals Main Activity Tips** - Lesson Recommendations
- **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.

- **Parameter** - An extra piece of information passed to a function to customize it for a specific need

Teaching Guide

Warm Up (15 min)

Introduction

Before class, set up a couple of paths in the classroom for students to walk on. Make sure the number of steps is obvious either using tape or cut outs of footprints. These paths should vary in length.

Gather the class together and note that there are different paths to walk down, but you don't want to have to write separate functions to walk down each one.

Instead, on an display the class can see, write or display the following

FUNCTION - "path", **PARAMETER** - "step"

- repeat "step" times:
  - walk forward

Ask the class if they know what the code you wrote means. Tell the class that instead of writing a unique function for each path, you wrote a function that can be customized to the length of the path.

This was done by declaring a function, "path", then giving it a parameter, "step". The variable, "step", can be used to hold the number of steps for each path.

Play with the function for each path, having a volunteer name the number of steps in a path and another volunteer walking down the path according to the code.

Main Activity (30 min)

CSF Express Course - Website

As the class works through these puzzles walk around and ask the following questions to each student.

- Are you using a function? Why or why not?
- If you aren't using a function, do you think a function could be helpful here?
- If you are using a function, are you using a parameter? Why or why not?
- If you aren't using a parameter do you think a parameter could be helpful here? Why or why not?

Sometimes the students won't need to use a function or a function with parameters, but they should always know**why** they are doing what they are doing.

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?
- Do you think parameters are helpful in code?
- When did you use a parameter and how did it change the way you wrote the rest of your program?

Standards Alignment

CSTA K-12 Computer Science Standards

- **AP** - Algorithms & Programming

Lesson 31: 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 Play 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, Play 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

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 either in **Course E 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)

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 32: 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 Play 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

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 33: 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 Play Lab or Artist as their programming environment. Now the students will be given their own space to create their project with either Artist or Play Lab. This will be the longest portion of the project. 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

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 34: 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 Play 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

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 35: 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 Play 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

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