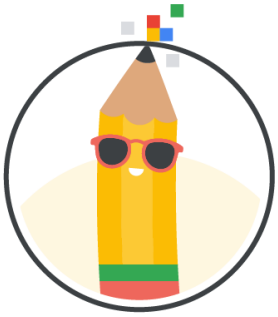
Lesson Plan



CS First Unplugged Lesson Plan

In This Lesson Plan

Additional Resources

CS First Unplugged Main Page
CS First Unplugged Starter Guide
CS First Unplugged Lesson Plan ← you are here
Activity Booklet (black and white)
Activity Booklet (color)

What is CS First?

Google CS First is a free computer science curriculum that makes coding easy to teach and fun to learn. CS First empowers every teacher to teach computer science with free tools and resources - no CS experience required. Sign in for free to get more benefits. (Not required for this activity)

In CS First, students code with Scratch, a free, block-based programming tool developed by MIT. It's ideal for beginners to learn coding. Students "snap" together blocks of commands that the computer can carry out.

Network a neighborhood and *Encode an emoji* activities were adapted from CS Unplugged materials, which are made available at csunplugged.org under a Creative Commons Attribution-ShareAlike 4.0 International license.

Overview

This lesson is a set of three *unplugged* activities, each of the which has been designed to introduce students to a key Computer Science concept without the need for a computer.

Tip: To extend this lesson, we've created a "plugged in" version of each activity in Scratch. Use these Scratch projects to help students see how their offline experience with a concept appears in a Scratch program.

The activities in this lesson can be completed individually and in any order. They cover a range of different CS topics and you can build student agency into the lesson by allowing students to focus on the activities they are most interested in. Additionally these activities have been designed to support students who are learning at home, including recommendations for three different implementations:

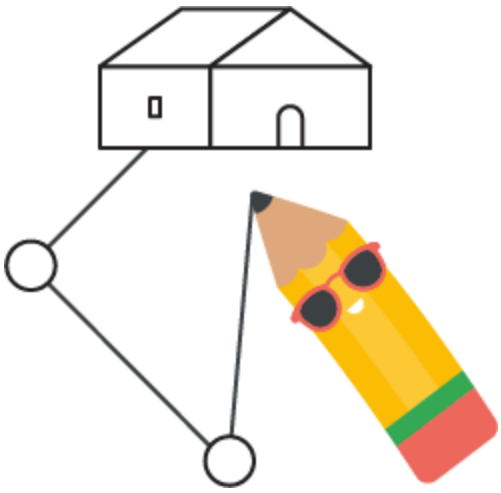
Remote - Each activity has been optimized for students working independently at home.
Blended - For classrooms meeting together virtually we provide some opportunities for students to collaborate.
In-Person - Classes meeting together in-person also have some opportunities for students to work together.

Materials

In addition to the activity booklet, some activities require or optionally benefit from additional materials.

- Small counters (like dried beans) to use on the Network a Neighborhood map.
- Scissors to cut out the Send a Secret Message cipher wheel.
- Cardboard and glue to provide additional stiffness to the Send a Secret Message cipher wheel.
- A thumbtack, toothpick, or straightened paper clip to connect the Send a Secret Message cipher wheel.

Activity 1: Network a neighborhood



In this activity, students are provided with the map of a town, and they need to connect all of the buildings as efficiently as possible. While it is pretty simple to build a path that connects all of the buildings, students will need to be methodical to ensure that their path is as cost-effective as possible.

This is an example of a computationally hard problem. As the map of a city gets more complex, it requires exponentially more time to check for all of the potential solutions and make sure that you've found the best one.

Learning Objectives

Students will:

- Design a solution (graph) that connects all nodes (houses)
- Iterate on their solution to make it as efficient as possible
- Describe the process of solving this map as an algorithm

Computer Science Topics

- **Algorithm**: Step-by-step instructions to be executed by a computer
- **Heuristic**: A problem solving approach to find a "good enough" solution when finding the perfect solution is impractical or impossible.
- **Network**: A group of computers connected together.

Procedure

<table>
<tr><th>Remote</th><th colspan="2">Blended</th><th>In-person</th></tr>
<tr><td>Have students read the instructions in the activity booklet.</td><td colspan="3">Display the map for everyone, or direct students to look at on their own booklets.
[say] In this activity, you will help connect everyone in this small town to the internet. You can do that by filling in the spaces that connect each house, building a network that links every building together. You want to do this using as few spaces as possible, so think closely about ways to make your network as small as possible. Feel free to try out lots of different approaches to find the one that is the best.</td></tr>
<tr><td>Allow students to work through their solutions at their own pace.</td><td colspan="3">Set a 10 minute timer for students to come up with the best solution that they can.</td></tr>
<tr><td rowspan="3">Have students reflect on their process using prompts provided in the booklet.</td><td colspan="2">Ask a student to share their solution with the class. After that student has finished, ask for another volunteer that solved it a different way. Try to get as many different approaches as possible.</td><td>Pair students up and have them explain how they solved the problem to each other. Ask a few pairs to share their solutions with the whole class. Try to get as many different approaches as possible.</td></tr>
<tr><td colspan="3">Discuss how the approaches students described would work if the map were twice as big, and then 10 times as big. How would you know that you had the best solution when there are so many to check?</td></tr>
<tr><td colspan="3">Introduce the term heuristic as a kind of algorithm that gives us a "good enough" solution to a complicated problem. Ask students to brainstorm a heuristic for finding a "good enough" short path.</td></tr>
<tr><td colspan="4">(Optional) Have students visit the Scratch version of this activity to practice on some randomly generated maps.</td></tr>
<tr><td colspan="4">(Optional) The original CS Unplugged activity Muddy City includes more detailed information about related CS concepts including Minimal Spanning Trees and algorithmic efficiency. Consider having students come up with a heuristic for finding a path through the map and then try it out with Muddy City.</td></tr>
</table>

Activity 2: Encode an emoji

In this activity, students learn about a type of encoding that lets them reproduce 8x8 pixel black and white images using a relatively small amount of data. This simple encoding is an example of *lossless compression*, a technique to compress or reduce the amount of space something takes up in a computer without losing any information required to reproduce. This is in contrast to *lossy compression* which can often save space by making something even smaller, but you lose some information about the thing you're compressing.

Learning Objectives

Students will:

- Convert an encoded image into a drawn pixel image
- Design and encode new images
- Propose ways to extend the encoding for color images

Computer Science Topics

- **Encoding**: A way of representing information in a way that can be stored, transmitted, and reproduced by a computer.
- **Compression**: A technique to reduce the amount of space that data takes up in a computer.
- **Pixels**: Tiny individual squares that together make up computer screens.

Collaborating Remotely

If your students are working together over a video conferencing platform you can use the chat functionality for students to share their encodings. Either in a whole class session or in breakout rooms, have students type their encoded images into the chat. Other students can then attempt to recreate those encoded images using the blank grids in the booklet, graph paper, or hand-drawn grids on scrap paper.

This same approach also works well with the third activity *Send a secret message*.

Procedure

<table>
<tr><th>Remote</th><th colspan="2">Blended</th><th>In-person</th></tr>
<tr><td>Have students read the instructions in the activity booklet.</td><td colspan="2">(Optional) Introduce the activity with the Making Contact video from the original CS Unplugged Activity that this activity was adapted from.

Have students look at the pixel image of a house in their activity book. Ask them to consider different ways that they could describe that picture over the phone so that someone else could recreate it. After students have had some time to think, have a few share their ideas with the class.</td></tr>
<tr><td rowspan="3">Allow students to work through their solutions at their own pace.</td><td colspan="2">Let students complete the drawings labeled 1-6 independently. Once everyone has finished them, ask a few students to share their drawings. If students completed the drawing differently, have them walk through the process of translating the encoding to the drawing to “find the bug.”</td></tr>
<tr><td colspan="2">Give students 10 minutes to create and encode some new drawings of their own using the blank grids.</td></tr>
<tr><td>Ask for a volunteer to share their encoding with the class. While the student reads out their encoding, have the rest of the class attempt to recreate the image.</td><td>Pair students together and have one student read out their encoding while the other student attempts to recreate the image.</td></tr>
<tr><td>Have students reflect on their process using prompts provided in the booklet.</td><td colspan="2">Prompt the class to consider how they might encode a color image. After students have had some time to think and shair with a partner, discuss some of the ideas as a whole class. If you have time, allow students to try and create their own encoding system for color images.</td></tr>
<tr><td colspan="4">(Optional) Students can use this Scratch app to easily create more encoded images to share with each other and this Scratch app to draw their images on screen.</td></tr>
<tr><td colspan="4">(Optional) You can provide students with larger and more detailed drawings from the original CS Unplugged activity.</td></tr>
</table>

Activity 3: Send a secret message

In this activity, students explore how to keep data secure online by using a simple substitution cipher. This is a method famously used by Julius Caesar in his private communications. While this kind of encryption is easily broken, and therefore not used for anything serious, it demonstrates how a secret key can be used to transmit messages in a way that makes them difficult to read by anyone but the intended recipient.

Learning Objectives

Students will:

- Encrypt a message using a simple substitution cipher
- Decrypt a message using a simple substitution cipher

Computer Science Topics

- **Encryption** - a system to hide information in plain sight.
- **Decryption** - a system to change encrypted data back into a readable format.
- **Key** - the secret data needed to decrypt an encrypted message.
- **Cipher** - an algorithm for encrypting and decrypting messages.

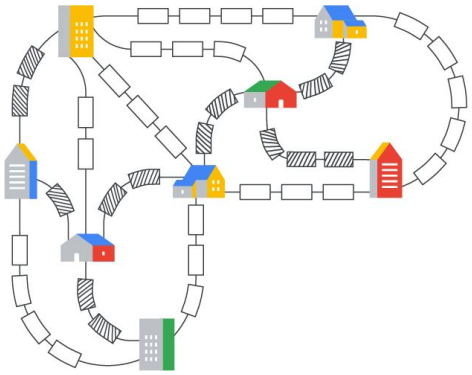
Procedure

<table>
<tr><th>Remote</th><th>Blended</th><th>In-person</th></tr>
<tr><td>Have students read the instructions in the activity booklet.</td><td colspan="2">Ask students to make a list in their head of all of the things that they have typed into a computer in the last week. Everything from passwords to login, to posts on social media or documents for schoolwork. Without sharing what is in their lists, ask students to discuss how many of those things they would prefer be kept private.

[say] The more things we use computers for, the more often we find things that need to be kept secure, which is why computer scientists have to think about encrypting information. But sending secret messages isn't anything new, and in fact today we're going to use a technique that is thousands of years old.</td></tr>
<tr><td>Allow students to work through their solutions at their own pace.</td><td colspan="2">Go through the first message in the activity together as a class before letting students build their cipher wheels and work on decrypting the rest of the messages.</td></tr>
<tr><td>Have students reflect on their process using prompts provided in the booklet.</td><td>Send new encoded messages through the chat and have students try to decode them.</td><td>Pair students up to take turns writing and decoding each others' messages.</td></tr>
<tr><td colspan="3">(Optional) Have students visit the Scratch version of this activity to encrypt and decrypt more messages.</td></tr>
</table>

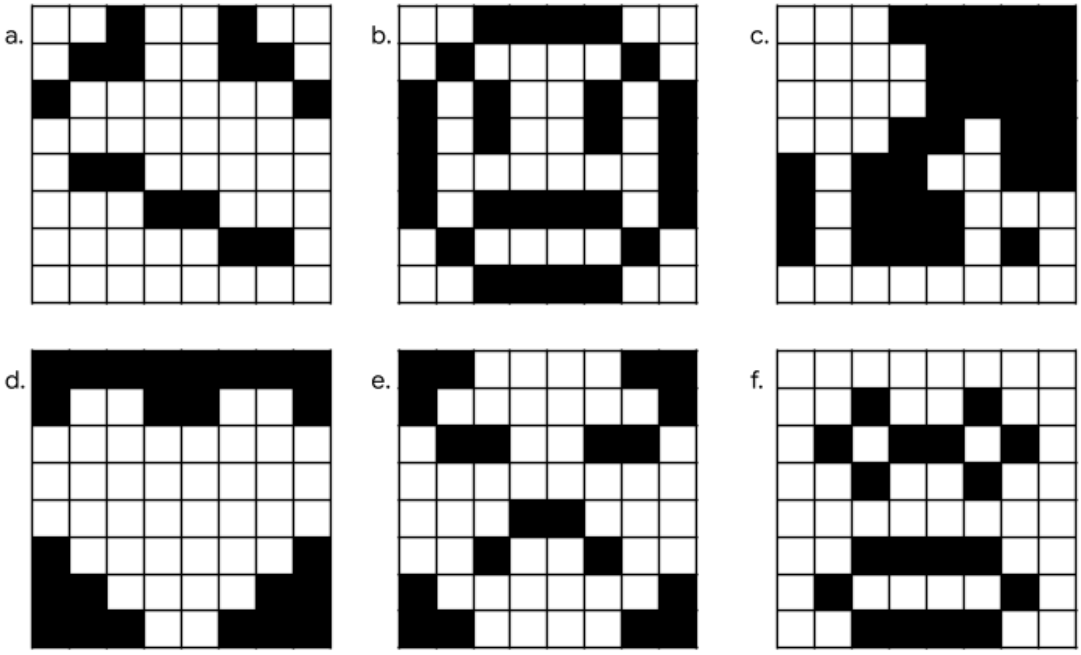
Answer Keys

Network a neighborhood answer key



There are lots and lots of different solutions to this problem (which is the whole point). This is one of many possible optimal solutions with a total length of 14 spaces. Encourage students to share and compare their solutions with each other to see how many different ways they can connect the buildings.

Encode an emoji answer key



Send a secret message answer key

Key	Encrypted Message	Decrypted Message
6	lujotm oy iuur	coding is cool
13	v ybir gb uryc crbcyr	i love to help people
23	jv ald fp qeb ybpq	my dog is the best
10	s dkvu dy bylydc	i talk to robots
3	orrn zkdw l ghfrghg	look what i decoded