## MOTIVATION

We've learned about encoding and decoding information, and we know that computers represent messages in binary. These messages can contain things like *data* or *instructions*. Instructions are essential to most of the computer programs we run: when we run a word processing program, the computer follows instructions to make a letter appear on the page when we press a key. When we play a game, the computer follows instructions to create the appropriate effect when we click on an object. These instructions are created by a computer scientist or software engineer, and they tell the computer what to do when it receives a signal.

This week, we'll encode an "instruction table" for building a LEGO tower using binary. We will need to break down the task into steps, describe those steps in a consistent and systematic way, and use our encoding skills to convert ideas into binary. Although our instruction table will not be the same as a computer program, the skills used to create the table are some of the skills that real computer scientists and software engineers use when they make the programs that we use every day.

## OBJECTIVES

Students will:

* Practice representing numbers in binary
* Use binary digits to encode and decode messages
* Understand the importance of binary numbers in computer science

## 

## RESOURCES

Lesson Plan (2 days): ([Source](http://www.muddx.com/c4x/HMC/MyCS/asset/LegoDataRepresentation.docx) / [PDF](http://www.muddx.com/c4x/HMC/MyCS/asset/LegoDataRepresentation.pdf))

Day 1 Activity: Encoding LEGO instructions ([Source](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1UnderstandingEncodingandDecoding.pptx) / [PDF](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1UnderstandingEncodingandDecoding.pdf))

Day 1 Activity Solutions ([Source](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1UnderstandingEncodingandDecodingSOLUTIONS.pptx) / [PDF](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1UnderstandingEncodingandDecodingSOLUTIONS.pdf))

Day 1 BONUS Activity: Additional LEGO Encoding Practice ([Source](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1AdditionalPractice.pptx) / [PDF](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1AdditionalPractice.pdf))

Day 1 BONUS Activity Solutions ([Source](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1AdditionalPracticeSOLUTIONS.pptx) / [PDF](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1AdditionalPracticeSOLUTIONS.pdf))

Day 2 Worksheet: Secret Fortress Construction Challenge ( [Source](http://www.muddx.com/c4x/HMC/MyCS/asset/Secret_Fortress_Construction_Challenge.pptx) / [PDF](http://www.muddx.com/c4x/HMC/MyCS/asset/Secret_Fortress_Construction_Challenge.pdf) )

# Lego Data Representation

## OBJECTIVES

Today we're going to learn how to encode a LEGO structure using binary numbers. Then we'll decode the binary numbers to create new structures.

## SAMPLE AGENDA

* Watch the "World Record LEGO Tower" video.
* Discuss the motivation: Why are instructions helpful?
* Agent Zhivago video
* Activity: Encoding LEGO instructions ([Source](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1UnderstandingEncodingandDecoding.pptx) / [PDF](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1UnderstandingEncodingandDecoding.pdf))
  + Discussion: Describing Instructions
  + LEGO Block Attributes video
  + DEMO: Building the Encoding
* Wrap-up video
* BONUS: LEGO Encoding Additional Practice ([Source](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1AdditionalPractice.pptx) / [PDF](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingDay1AdditionalPractice.pdf))
* Discussion: binary encoding debrief

## VIDEO: MOTIVATION: WORLD RECORD LEGO TOWER

## MOTIVATION: CONTEXT

This video showed the construction of the tallest LEGO Tower ever built. It involved teamwork and collaboration from families and schools all over Brazil, and probably a lot of organization!

**How would you organize a big project like this?**

Would you tell people what to do? Give an example of an instruction you might give.

What's difficult about giving instructions? What's difficult about following instructions?

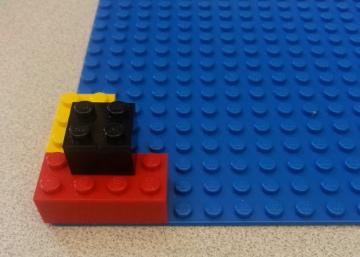
Is giving instructions to a computer different from giving instructions to a human?

## VIDEO: AGENT ZHIVAGO

## ACTIVITY: ENCODING LEGO INSTRUCTIONS

Agent Zhivago is trapped and can only escape if he builds a LEGO key. However, the instructions for building the key are *encrypted* in some kind of *code* that he has to *decode*. In fact, Agent Zhivago has recognized the code as *binary*, which is a code you've been working with.

But before we look at coded instructions, let's think about instructions in general. Suppose we built this LEGO tower:



**How would you tell a friend how to build the exact same tower?**

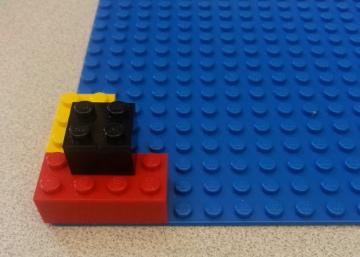
Write down a set of instructions on the activity handout:



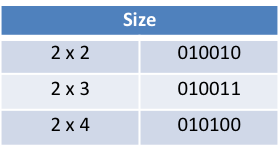
## UNDERSTANDING BLOCK ENCODING AND DECODING

LEGO blocks have four attributes: color, size, position, and orientation. These attributes can help us when we make instructions!

**How would you describe the four attributes of each block in this tower?**

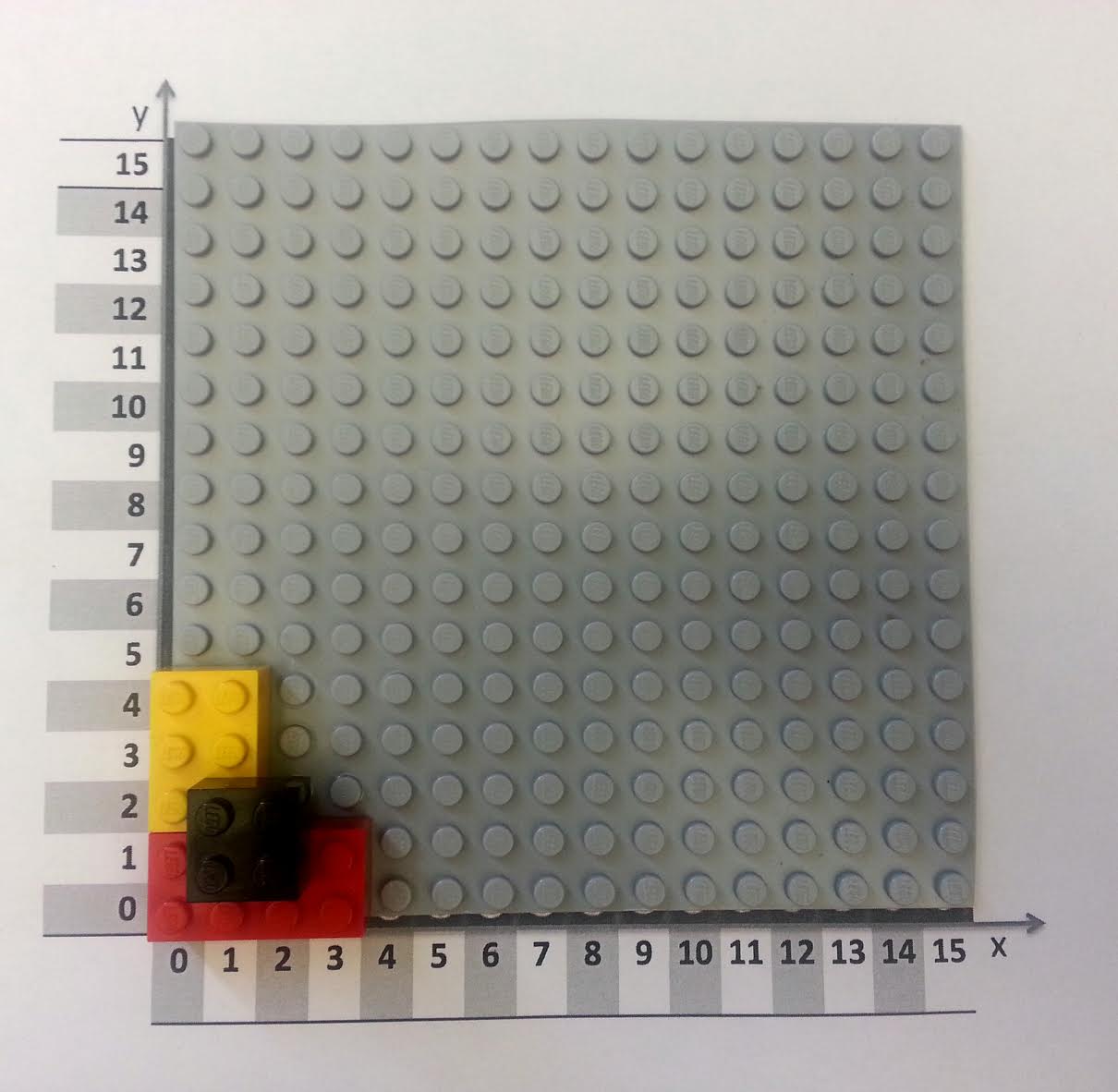


We want to be able to describe these attributes with binary, so let's *encode* the four attributes. First, let's look at color and size. We can create the following codes:

**What kind of block is represented with color 010 and size 010010?**

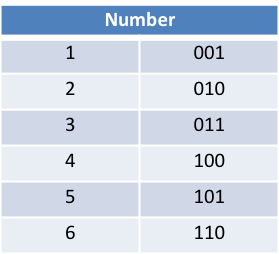
Next, we'll have to define a way to represent position. We want this to be as clear and simple as possible, so we define a coordinate system:



The position of each block can be defined by the x and y coordinates of its bottom left corner.

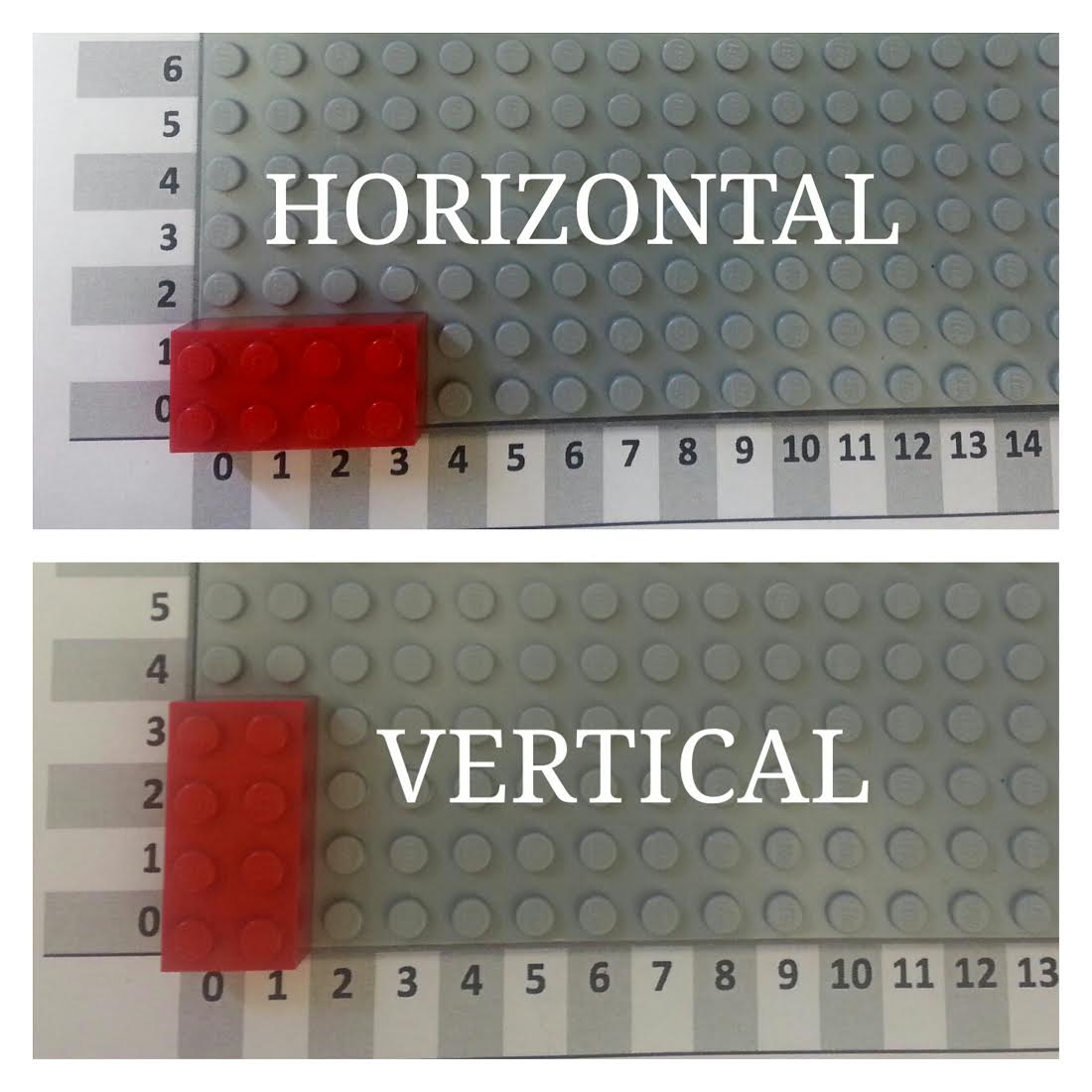
**What are the coordinates of each of the blocks?**

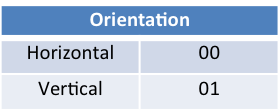
Then, we can just convert these coordinate numbers into binary:



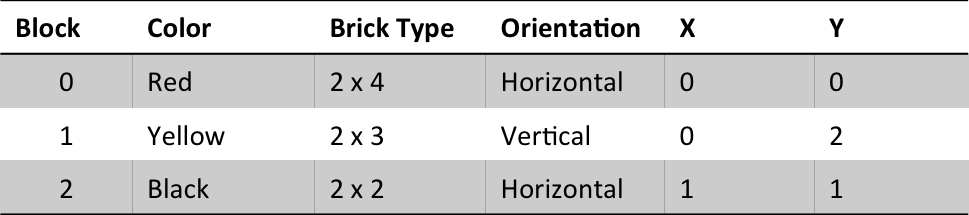
These are the same binary number conversions you've been practicing!

Last, we can assign binary codes for orientation. Here, we're mostly worried about which way the block is rotated:

 We can encode these orientations:

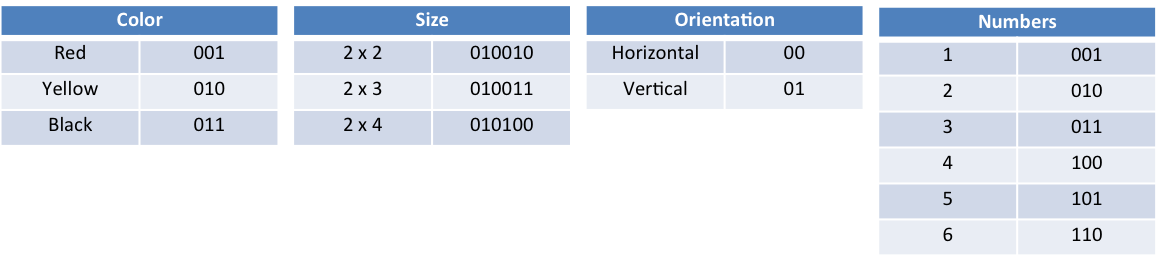


We've described each LEGO block's four attributes (color, size, position, and orientation) in normal language and in binary. Now, instead of giving the instructions in sentences, we can put it into a table where each line represents a step of the instructions. Notice that we've numbered the blocks, so that there's one step for each LEGO block.

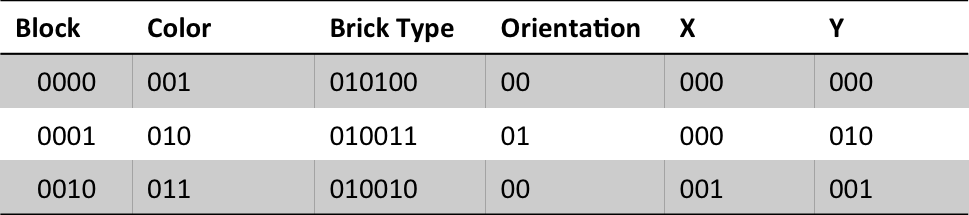


**Can you convert the information in this table to sentences to tell a friend how to build the tower?**

Using our code, we can translate all of the information in this table into *binary*. Here's the legend for our code:



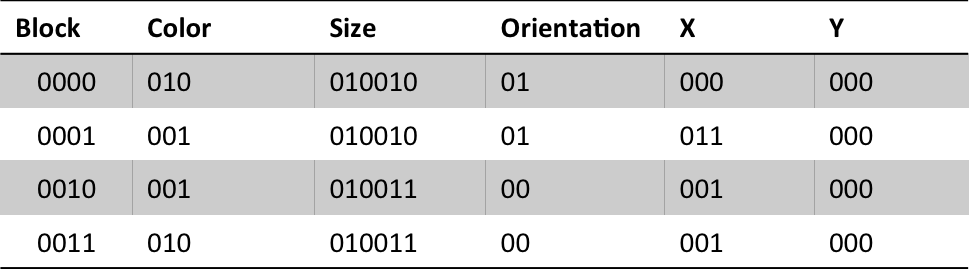
And here's the binary instruction table:



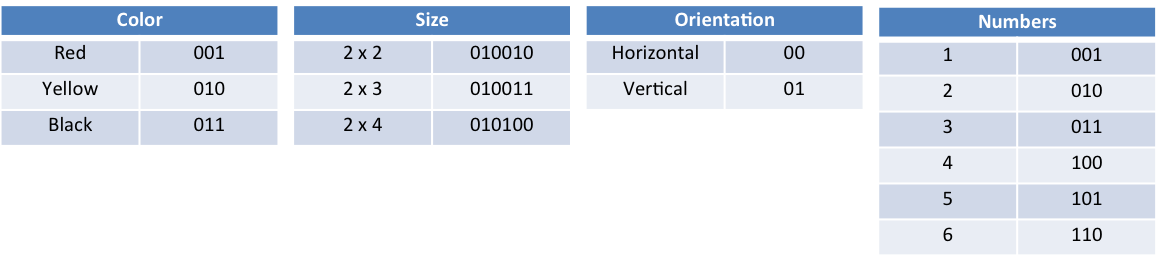
**How would you use the binary instruction table to build a structure? Would you need to *decode* it first?**

## ACTIVITY: BUILDING THE ENCODING

Here's an encoded instruction table for a small LEGO tower. Can you build the tower?



Here's the legend for our encoding:



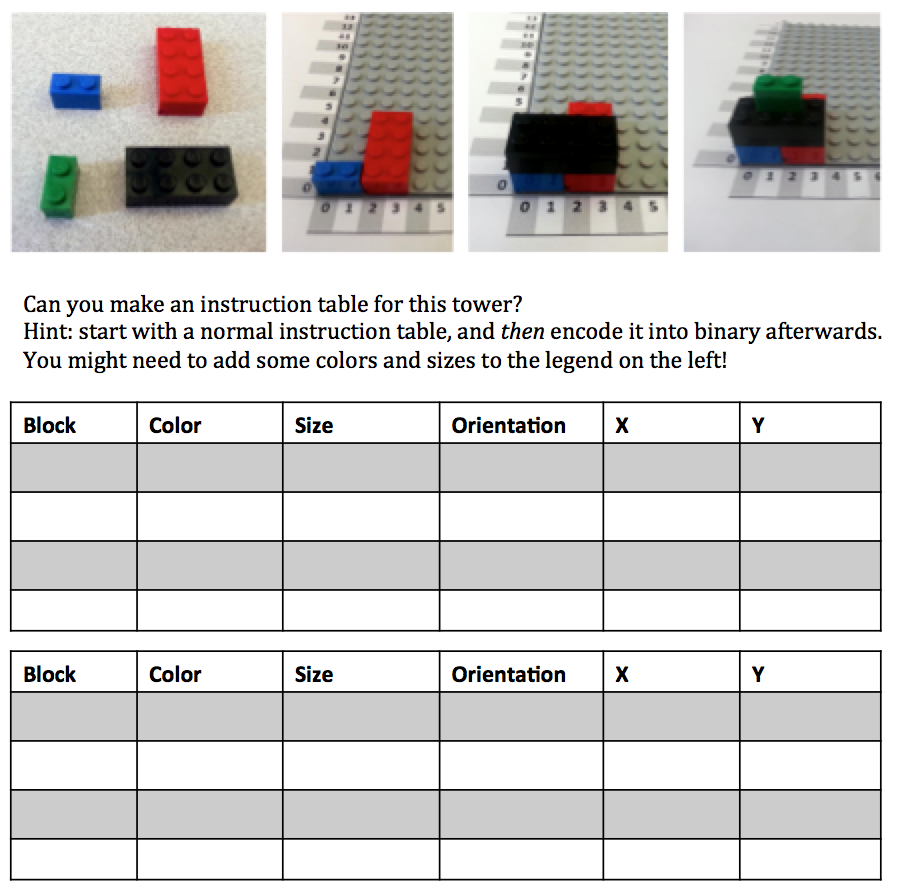
Don't have LEGOs on hand? No worries! You can build the structure virtually using [Build with Chrome](http://www.buildwithchrome.com/static/builder) online.

If you need a hint, click [here](http://www.muddx.com/c4x/HMC/MyCS/asset/LEGOEncodingActivityOriginalTable.png) for the decoded instruction table.

## VIDEO: ACTIVITY WRAP-UP

## BONUS: PRACTICE ENCODING/DECODING LEGO INSTRUCTIONS

Want or need more practice encoding and decoding LEGO instructions? Try these exercises!



## DISCUSSION

Today, we created a binary encoding for LEGO blocks so that we could use binary to express the instructions for building a LEGO tower.

**What are the advantages of encoding instructions in binary? What are the disadvantages?**

How could you fix some of those disadvantages?

**What kinds of encoding and decoding skills did you use?**

What steps were involved in creating the binary instruction table? What steps were involved in *decoding* a binary instruction table?

Did you use binary encoding the same way as you did in the last unit, or did you use it in a different way? How?

What was difficult about encoding instructions in binary?

# Lego Data Representation II

## OBJECTIVES

Today we will use our binary encoding and decoding skills to encode our own LEGO structures with binary numbers. Then we'll try to replicate a structure from a binary instruction table. We'll see that precision and accuracy are key in creating replicable instructions!

## SAMPLE AGENDA

* Discuss the motivation and context: Why are instructions so important? How must instructions be expressed to most computers?
* Activity: Secret Fortress Construction Challenge
  + Watch the Introduction video
  + Recall the tallest LEGO tower video from Day 1
  + Complete the activity and worksheet ([Source](http://www.muddx.com/c4x/HMC/MyCS/asset/Secret_Fortress_Construction_Challenge.pptx) / [PDF](http://www.muddx.com/c4x/HMC/MyCS/asset/SECRET_FORTRESS_CONSTRUCTION_CHALLENGE.pdf))
  + Watch the Activity Wrap-Up video
  + Reflect: How did things go? What skills did we use?
* Watch the Career Connection: 3-D Printing video to conclude the class

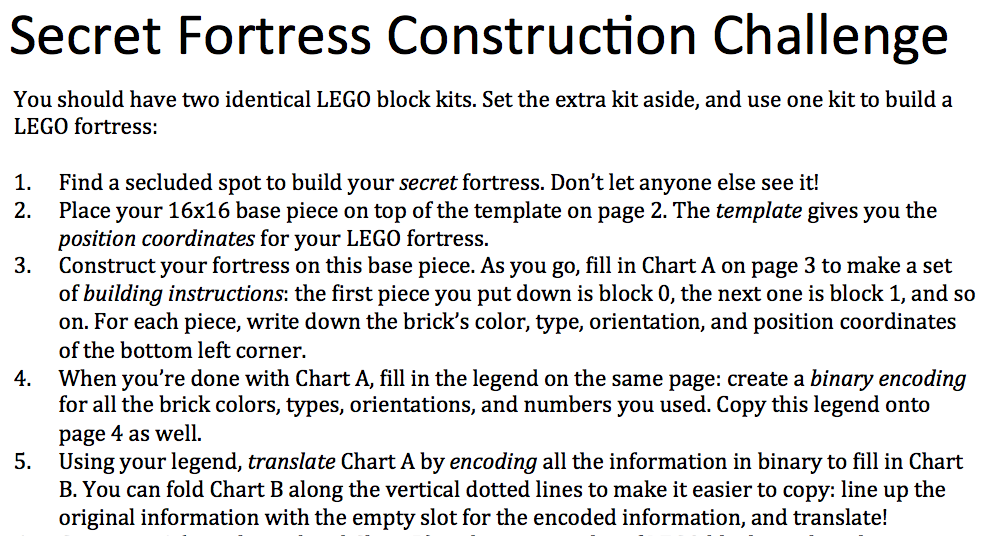
## VIDEO: SECRET FORTRESS CONSTRUCTION CHALLENGE INSTRUCTIONS

## ACTIVITY: SECRET FORTRESS CONSTRUCTION CHALLENGE

Remember this tower? Today you'll build your own LEGO fortress.



Start with a LEGO building kit (9 LEGO blocks and a base piece) and the Secret Fortress Activity Handout (linked below).



Without letting other groups see, build a fortress and create a set of building instructions using the first chart in the packet. Then, create a legend to convert your instructions into binarycode. Using this legend, you'll be able to create an instruction table in binary. When you finish your instruction tables, swap your binary instructions with another group and try to build each others' fortress. Did it work?

## VIDEO: ACTIVITY WRAP-UP

## REFLECTION



**Was any team able to replicate the orignal fortress *exactly*? How many were close?**

If not, what was wrong with their structure? Wrong order? Wrong orientation?

Was there a fault on the original side in encoding? Or was it a fault on the decoders' side in decoding the information?

Were any of the instructions ambiguous?

**What skills did you use to create a binary instruction table? What skills did you use to build a structure from binary instructions?**

What was the hardest part? Why was it hard?

The process of creating instructions based on an idea and converting these English instructions to binary instructions is common in computer science. When a computer scientist is given an idea for a program, he or she starts by figuring out what task the program should accomplish. Then, he or she breaks the task down into a series of steps and create instructions to carry out each task. Last, these instructions are encoded into some kind of language a computer can understand, so that the computer can carry out these tasks!

Does this sound a little like what you just did? It should!

## VIDEO: 3-D PRINTING