

Helping Kids Learn – Post #7 4/15/20


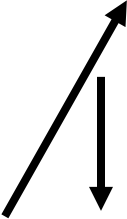
Note: There are two versions of this Investigation so 2 players can use different codes. This is Version 2.

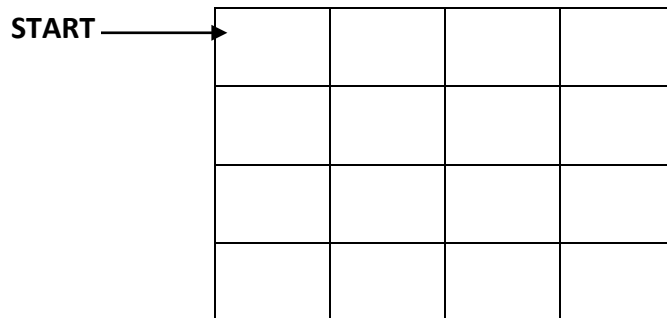
# STEM: Technology – Coding

Every good spy story involves a code. A code is simply a way to communicate information. Although we don't think of them this way, letters of the alphabet are a code. That's why we have to *learn* to read, usually by first learning the sound to make when we see symbols like **a,b,c**. To make the code sneaky and complicated, some letters, **a** and **c** are good examples, make more than one sound! In this investigation, we look at computer coding, the basis for much of our communication today. Any age can learn to code. See [Lift the Level](#) below for suggestions.

## Be a Programmer – Creating and Writing Code

Follow these directions on the grid from top to bottom.







 <p>Start at the top left corner. Move one square to your right. Draw an <b>X</b> in the square. Move one square to your right. Draw an <b>X</b> in the square. Move one square to your right.</p>		<p>Move one square down. Draw an <b>X</b> in the square. Move one square to your left. Move one square to your left. Move one square down. Draw an <b>X</b> in the square.</p>
---	--	--



You just followed **code** to move on a **grid** and draw **X**s. (You can check your work with the grid on p. 4.)

### *Too many words!! There's got to be an easy way!*

There is! Use **symbols** to show what to do:

-  Start at the top left corner.
-  Move one square to your right
-  Draw an **X** in the square
-  Move one square to your left
-  Move one square down
-  Move one square up

Try this one:

★  
×  
→  
↓  
×  
→  
↓  
×


You can write the symbols from left to right instead:

★ × → ↓ × → ↓ ×

Your finished grid should look just like the first one, see p. 4.

Work with a partner. **DON'T SHOW THEM YOUR CODE!**  
Ask your partner to follow your directions on their grid.

Read the code like this:  
*star x down arrow right arrow right arrow ...*

Your code: ★ × ↓ → → × ↓ × ↓ → → ×

**Answer key for your code:**

×				
		×		
		×		
				×

**Follow your partner's code:**


Then you switch – follow your partner's directions on the grid above.

Check each other's grids. Fix any mistakes.

If you are not playing with a partner, the answer key is on p. 4. Work backwards: write the code that gets this answer!

Now write your own code. **DON'T SHOW YOUR PARTNER!** Put ✕s where you want in this grid.


Write code in the box below to show your partner what to do to make the same grid design. Fold your paper on the - - - - line or cover the screen so your partner can't see your grid. Trade and follow the code. Check your work with the grids you both made.

Your code:

**Lift the Level** You can make this lesson deeper and/or suitable for different learners by any of the following:

1. Use drawing symbols other than ✕.
2. Use more than one drawing symbol.
3. Make additional rules (programmers call them *constraints*) such as you can't land in the same grid square more than once.
4. Music written on paper is a complex code telling the viewer/musician the sound (pitch), length of sound, and style (slurred, connected, staccato, etc.). Write or play some music from the point of view of a programmer.
5. There are many, many alphabets. Invent your own. Be sure to include how to pronounce (what sound to make) each symbol. Don't forget money (currency) symbols like \$, €, £, ¢, ₤.
6. Math has a whole range of symbols, some of which are used in programming. Make a *glossary* to explain math symbols.
7. Explore a programming language such as HTML (used to create this website, for example). Or Python, used in many student applications. See [STEM Online](#) below.
8. Learn to find and fix coding errors by playing Help Gidget. See [STEM Online](#) below.

## STEM Online

These are all really cool sites!

**HTML Programming Language** See the online HTML help from the group that develops web programming standards at <https://www.w3schools.com/>.

**Python Programming Language** Many student applications are written in Python including many robotic applications. There's a great tutorial at <https://projects.raspberrypi.org/en/projects/turtle-amazing>.

**Debugging Programs** Most of us make mistakes when we write. We cross out or erase and make changes as we go. Programmers/coders make mistakes, too. In hundreds of strings of code, it can be very hard to find one single missed letter or symbol. So, *debugging* a program can take even more time than writing it. This free, online game teaches debugging in a fun way. It was developed by NJIT Professor Michael Lee and others with lots of help from middle school students. [www.Helpgidget.org](http://www.Helpgidget.org)

## NJ Student Learning Standards

### 8.1 Educational Technology

[International Society for Technology in Education \(ISTE\) Standards for Student](#)

[American Association of School Librarians \(AASL\) Student Standards for the 21st-Century Learner](#)

[Common Sense Student Standards Alignment in the K-12 Digital Citizenship Curriculum](#)

### 8.2 Technology Education, Engineering, Design and Computational Thinking - Programming

[K12 Computer Science Student Framework Statements by Grade Band](#)

[International Technology and Engineering Educators Association Standards for Technological Literacy](#)

## Solutions

Page 1

	×	×	
			×
	×		

Page 2 – What your answer should look like:

		×		
		×		
		×		
		×	×	