



GEMS Challenges -- Programming Languages

SUBJECT

Computer Science

Suggested grades

2-8

Cautions/concerns

You can make this more difficult with harder drawings but start easy.

The Challenge—Understand the importance of clear communication in life and in programming**LEADER ACTIONS****GIRL ACTIONS**

INTRODUCTION	Present the challenge—how does a computer know what to do? It is just a machine!	Brainstorm options
BACKGROUND INFORMATION	Have the girls think about the kinds of things we are asking computers to do for us now—drive cars, fly planes, manage our money.	
ACTIVITY	<p>Before passing out any materials, do the demonstration example in the lesson plan on the board with one of the girls. I suggest that you ask the girl to do the drawing and you as leader give the instructions. This way, if she does not replicate the drawing well, you can take the blame for your poor instructions. You also can model hiding the drawing you are describing from the person who is following your instructions. Discuss how this went. Was it an accurate copy?</p> <p>Divide the girls into teams of two and have them sit across from each other, with enough space between them that the groups won't hear each other. Pass out pencil and paper to one girl (the 'computer') on each team, and a copy of figure A to the other girl (the 'programmer').</p> <p>Tell each team that the programmer is to give clear instructions to the computer on how to replicate the figure. The programmer must not look at the</p>	In teams, take turns describing how to draw the figures.

LEADER ACTIONS**GIRL ACTIONS**

	<p>computer or correct mistakes.</p> <p>When they have completed the first drawing, ask the teams to share. What worked? What didn't?</p> <p>Next, switch roles, pass out new supplies and attempt figure B. Complete all figures in this way.</p>	
CONCLUSION	<p>Talk with the girls about the process and level of accuracy they achieved. What helped? Did they get better as they practiced? Were there certain words that made the process easier? Would it have been easier if the teams had checked their work after each step?</p>	<p>Think and talk about the process.</p>
REFLECTION	<p>Ask 2-3 girls to draw a card and reflect on the experience.</p>	

Supplies

- Pencils
- Paper
- Copies of figures copied and cut into individual figures

Preparation needed

Photocopy page two of the lesson from csunplugged – one for each team, cut the figures apart and sort them into bags—a bag of figure A, a bag of B, etc. This way no one sees the figures ahead of time as they get harder.

Comments

Adapted from CSunplugged.org

Activity 12

Marching Orders—*Programming Languages*

Summary

Computers are usually programmed using a “language,” which is a limited vocabulary of instructions that can be obeyed. One of the most frustrating things about programming is that computers always obey the instructions to the letter, even if they produce a crazy result. This activity gives children some experience with this aspect of programming.

Curriculum Links

- ✓ English: Interpersonal Listening Level 3

Skills

- ✓ Giving and following instructions.

Ages

- ✓ 7 years and up

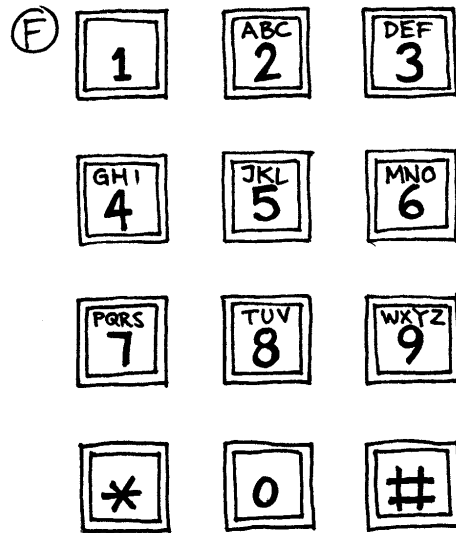
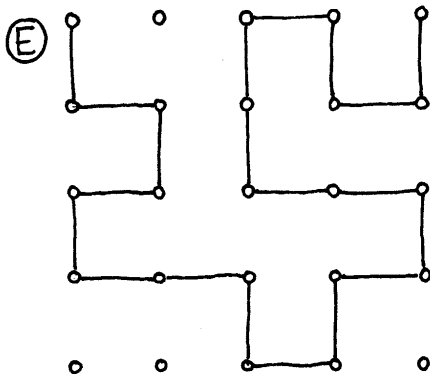
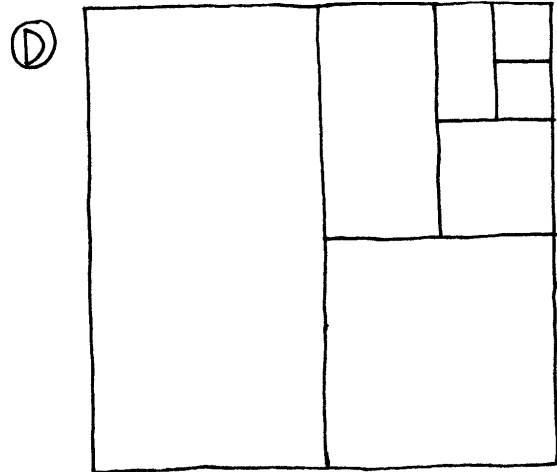
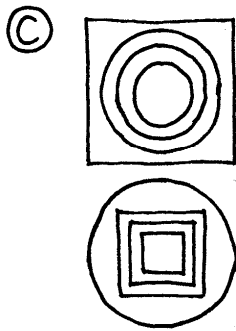
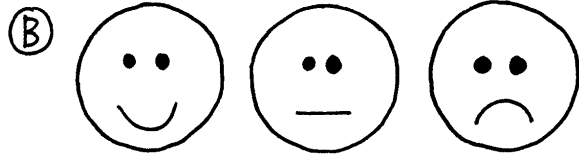
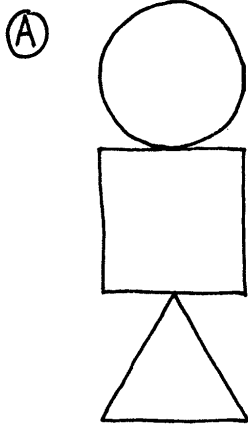
Materials

You will need:

- ✓ Cards with pictures such as the ones shown on the next page.

Each child will need:

- ✓ Pencil, paper and ruler



Marching Orders

Introduction

Discuss whether it would be good if people followed instructions exactly. For example, what would happen if you pointed to a closed door and said, “Go through that door”?

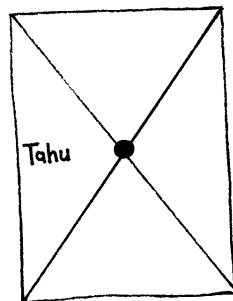
Computers work by following lists of instructions, and they do exactly what the instructions say—even if they don’t make sense!

Demonstration Example

See if the children can draw the picture from these instructions.

1. Draw a dot in the centre of your page.
2. Starting at the top left-hand corner of the page rule a straight line through the dot finishing at the bottom right hand corner.
3. Starting at the bottom left-hand corner of the page rule a line through the dot, finishing at the top right hand corner.
4. Write your name in the triangle in the centre of the left-hand side of the page.

The result should look something like this:



Activities

Choose a child and give them an image (such as the examples on page 102). The child describes the picture for the class to reproduce. The children can ask questions to clarify the instructions. The object is to see how quickly and accurately the exercise can be completed.

Repeat the exercise, but this time the children are not allowed to ask questions. It is best to use a simpler image for this exercise, as the children can get lost very quickly.

Now try the exercise with the instructing child hidden behind a screen, without allowing any questions, so that the only communication is in the form of instructions.

Point out that this form of communication is most like the one that computer programmers experience when writing programs. They give a set of instructions to the computer, and don't find out the effect of the instructions until afterwards.

Have the children draw a picture and write down their own instructions. Try them out in pairs or as a whole class.

Variations

1. Write instructions to construct a paper dart.
2. Write instructions on how to get to a mystery location around the school using such instructions as "Go forward x metres", "turn left" (90 degrees), and "turn right" (90 degrees).

Children should test and refine their instructions until they have the desired effect.

3. Blind Game. Blindfold a child and have the other children direct them around the room.

What's it all about?

Computers operate by following a list of instructions, called a program, that has been written to carry out a particular task. Programs are written in languages that have been specially designed, with a limited set of instructions, to tell computers what to do. Some languages are more suitable for some purposes than others.

Regardless of what language they use, programmers must become adept at specifying *exactly* what they want the computer to do. Unlike human beings, a computer will carry out instructions to the letter even if they are patently ridiculous.

It is important that programs are well written. A small error can cause a lot of problems. Imagine the consequences of an error in the program of a computer in a space shuttle launch, a nuclear power plant, or the signals on a train track! Errors are commonly called “bugs” in honour (so it is said) of a moth that was once removed (“debugged”) from an electrical relay in an early 1940s electronic calculating machine.



The more complex the program, the more errors there are likely to be. This became a major issue when the USA was working on the Strategic Defence Initiative (“Star Wars”) program, a computer controlled system that was intended to form an impenetrable defence against nuclear attack. Some computer scientists claimed that it could never work because of the complexity and inherent unreliability of the software required. Software needs to be tested carefully to find as many bugs as possible, and it wouldn’t be feasible to test this system since one would have to fire missiles at the United States to be sure that it worked!