

Activity: Thinking Like A Computer // Code a Friend

Time: 45-90 minutes (optional advanced lessons)

Age: 6 (quite young) -- 12

Agenda:

5 min: warm up/brain teaser (whole class)

- Can you trace this shape without lifting up your pencil or drawing over any lines you've already drawn? (The line can pass through the points!)



10 min: brief lesson/activity introduction (whole class)

- What is a computer program?
- Testing for bugs
- Controlling the Robot / Explaining Activity Instructions
- How to give the Robot instructions

20-30 min: activity -- code a group member (groups of 4-5)

- teach them to walk
- take turns being the robot
- competition: robot dance-off

EXTRA 30 min -- loops and functions

- 10 min lesson: loops and functions (whole class)
- 20 min prep for competition: build and test your program (groups of 4-5)
- adding your own instructions
- competition: robot race: cross the classroom, pick up an item, come back

5-10 min: retrospective (whole class)

- Height vote on how we liked the activity
- What would we change about the activity for next time?
- What were some of the unexpected challenges we faced, and how did we overcome them?
- What was something our team did well together?
- What did you do when the Robot malfunctioned? How did you fix it?
- official retro activity? (Starfish? Air Balloon?)

Materials:

- 15 Pre-written Notecards per group (2 extra for Loop and Function for advanced)
(Assume 30-40 students, ~10 Groups!)
- Pen & Paper for drawing in warmup
- Extra notecards for individual functions

NOTES:

- This entire thing is a choose-your-own-adventure. I found an introduction to myself and robots, coding, what an interface is is a good start.
- It's good to preface with "this is engineering, so there are no wrong answers." Kids seem quite hesitant at first.
- Warm up could be Intro activity Explain "interfaces" and "instructions" as robots understand them -- the robot can only understand "up, down, left, right" for these. Have the children "control" you by placing your pen down (on a whiteboard works well) and following their instructions to trace over the shape. They'll see it's difficult, probably mess up, so then do the next one! It's easier. This is a good place to split up into groups/introduce the activity.
- This gets them out of their shell/answering questions
- Start by introducing the activity and having the students choose your
- For younger or inexperienced groups, the flash cards work extremely well. Older kids or kids with coding experience find them very boring.
- It works well to have one programmer and multiple robots per group. Keeps everybody occupied.
- Give older or more experienced students a challenge straight away. For example, ask them to get their robot (using primitives) to take a step.
 - This goes well into the function activity. Show how to make a function "Step," and parameterize it with "side" (L or R), then how to "walk." Often this involves revising the function on the board as a class.
 - Ask how they figured out how to do things (hint: break the problem down to see how humans do it)
 - Show how everybody solved the problem in slightly different ways.
- Key CS concepts:
 - Interfaces
 - Functions
 - Building on interfaces *using* functions to create higher-order behavior
 - Interface limitations

Good Robot Challenges:

- Stand up
- Walk
- Jump
- Introduce itself (handshake, fist bump, etc)
- Dance contest (they loved this. The internet is full of funny dances. Macarena works well)

Retrospective Questions:

- Did different robots follow instructions differently? How did we get around that?
- Did you ever encounter a time when instructions didn't make sense? What did you do?
- What obstacles did you run into (literally and figuratively)
- How did you problem solve? (hint: iteration, test as you go, try it out and tweak it)
- How did the way the robots behaved differ from how people would do things?

Programming Robots



You're a group of research scientists! One person in each group will start as the Robot, and another as the Programmer. Using the instructions below, all group members besides the Robot will "program" the Robot to move. The catch is, the Robot can only do EXACTLY what the instruction says.

- The person who had a birthday the most recently starts as the **Robot**.
- The person who has the longest hair starts as the **Programmer**.

How to Program the Robot:

To program the Robot, TOGETHER the group must decide what instruction they want to give. Then the Programmer then holds up those cards, and says the instruction.

Instructions consist of 2 things: 1 Action and 1 Target, 1 Modifier (optional).

The 7 Possible Actions are:

- Bend
- Straighten
- Lift
- Extend
- Open
- Close
- Lower

The 7 Possible Targets are:

- Waist
- Leg
- Knee
- Elbow
- Arm
- Hands
- Eyes

Modifiers:

- Right
- Left



So for example, an instruction could be "Lift your Right Leg" or "Bend your Left Elbow." Then, the Robot would perform that instruction.

The Rules:

- You can only give the Robot one instruction at a time.
- You cannot give the Robot another instruction until it has completed the first one.
- The Programmer can only talk to their group's Robot.
- Robots can only receive instructions from ONE Programmer.
- After 5 minutes, the Robot and the Programmer change

Challenges and Questions:

- See if you can get the Robot to pick something up off the floor.
- Try to get the Robot to draw a picture.
- What sort of funny dance can you get the robot to do?

Programming Robots -- Advanced



Now that you have an idea of how to program Robots, you may find you have to give many instructions over and over again to get them to do the smallest things! Instead, now you can use **Loops** and define **Functions** that are able to give the Robot multiple instructions at a time!

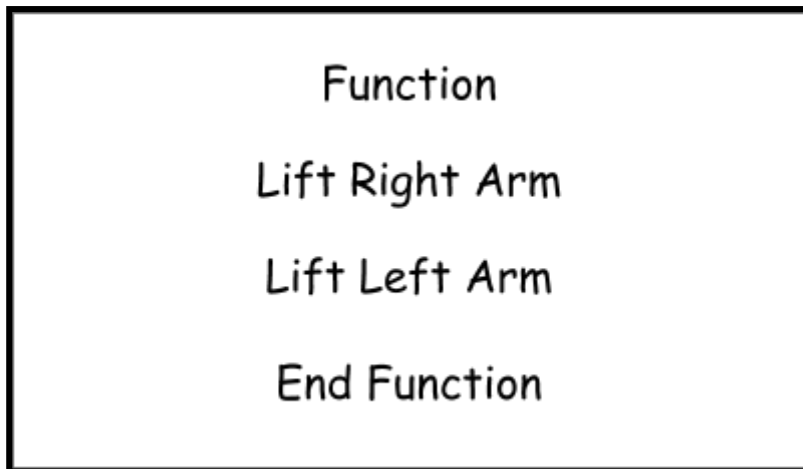
How To Use Functions:

Now you have a **Function** card. This allows you to group instructions together into easy-to-understand and commonly used chunks. To use this card, the Programmer **defines a function** by saying "Function," giving a name, listing *multiple* instructions, and then saying "End Function!"

After you define your Function, you can write those instructions on a new card with the title of the Function. From that point on, the Programmer can then say the name of the function, and the Robot performs all the instructions that were in the Function.

For example, I could say:

"Function Lift Both Arms, Lift Right Arm, Lift Left Arm, End Function!"



Now, whenever I want the Robot to raise both arms, I can just say "Lift Both Arms!"

Make sure that your Robot knows what instructions are in the Function! And, make sure when you use that function, the Robot is not doing any extra steps!

How To Use Loops:

You also have a **Loop** card. This gives you a third step in the instruction. Now, instead of just saying an Action and a Target, you can say a **number of times to repeat that instruction**. For example, I could say "Bend your Waist, 2 Times." This would make my Robot bend over twice!

Challenges:

- How would you create a function using instructions to program the Robot to take a step?

- What happens when you try to make the Robot do a Loop for things they already did? How can you fix it?
- How can you use Loops and Function at the same time to make the Robot walk? Be careful not to make the Robot run into something!