Lesson 2: The Need for Algorithms

ADAPTED FROM CODE.ORG CURRICULUM

Objectives: You will be able to…

- Trace programs written in the “Human Machine Language”
- Develop an algorithm to find the smallest playing card in a row of cards
- Express an algorithm in the “Human Machine Language”
- Identify the properties of sequencing, selection and iteration in the “Human Machine Language”
- Evaluate the correctness of algorithms expressed in the “Human Machine Language”

Getting Started

- Recall the lessons learned about language: Yesterday's activity focused on the inherent difficulties of trying to express precise processes with written language.
- A few key points:
  - We need to agree on a set of commands and exactly what terms mean
  - The fewer commands we have, the easier it is to agree
  - We want to know what are the "primitive" operations – the most basic set of operations that will allow us to do most of the tasks that the situation requires

Define: Algorithm

- The art (and science) of using a well-defined language of primitive operations to solve problems is the art and science of **algorithms**
- Algorithm – a precise sequence of instructions for processes that can be executed by a computer and are implemented using programming languages
  - See AP Computer Science Framework
  - Note: Sequencing, selection, and iteration are the building blocks of algorithms (you will know what these mean in the future)
- Note: this is one of the 7 "Big Ideas" for AP CSP

Define: Algorithm

- One way to think of the study of algorithms is that it is a study of processes – how can you use a small set of instructions to clearly and correctly define a process that will solve some problem?

- In the last lesson, you used LEGO blocks and you attempted to design and algorithm
  - Any time you are trying to write a precise set of instructions for a process to solve a problem you are designing an algorithm

Activity 1: Find Min Card Algorithm

- In Computer Science we are interested in computational processes – ones that can be executed by a computer – which have specific sets of constraints
- We are going to start by thinking of ourselves as a "Human Machine" that operates on playing cards on a table.
- We will use the "Minimum Card Algorithm – Activity Guide"

Activity 1: Find Min Card Algorithm

- We will use the "Minimum Card Algorithm – Activity Guide"
- Work in pairs and think about:
  - How do you know when to stop?
  - Do your instructions state where and how to start?
  - Is it clear where to put cards back down after you've picked them up?

Activity 1: Find Min Card Algorithm

- As we look at these algorithms you came up with, we can see some common things you are all making the human machine do and commonalities in your instructions
- Can we define a language of common Human Machine commands for moving cards around?
- What are the commands or actions most of these instructions have in common?

Activity 1: Find Min Card Algorithm

- What are the commands or actions most of these instructions have in common?
  - SHIFT – some form of shifting hands one position down to the row left or right
  - MOVE – some form of moving a hand directly to a particular card based on its position
  - COMPARE – some way to compare cards and do something based on the result
  - GO TO LINE – some way to jump to an earlier or later line in the program
  - PICK UP/PUT DOWN – when to put the card back

Activity 2: The "Human Machine" Language

- To be clear, let's formalize what we have been doing into a language…
- We are going to use the "Human Machine Language – Activity Guide"
- Step 1: Read the first page
- Step 2: With a partner figure out the example programs (one reads, one acts out)
- Step 3: Review as a class

Activity 2: The "Human Machine" Language

- Challenge: Find Min with the Human Machine Language
- First identify what's different about the problem setup for the Human Machine Language
- Second, use the Human Machine Language to write the algorithm for finding the min card
- Finally, share solutions with other pairs

Wrap-up: The "Art" of Programming

- Notice two things about algorithms and programming…
  - Different algorithms can be developed to solve the same problem (EK 4.1.1H)
  - Different code can be written to implement the same algorithm
- In programming, just like art, we strive to make beautiful things:
  - A beautiful algorithm is an elegant and clever idea for how to solve a problem
  - A beautiful program is an elegant use of whatever language structures are provided to make the algorithm actually work on a computer