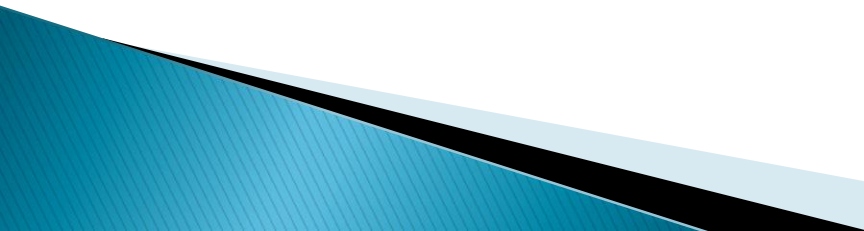


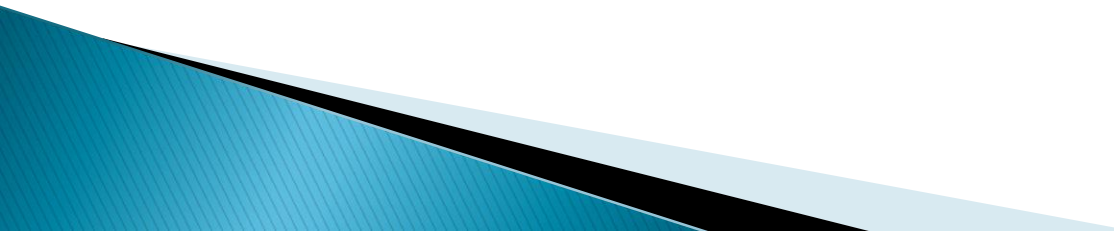
Lesson 3: Creativity in Algorithms

Adapted from code.org curriculum

Objectives: You will be able too...

- ▶ Develop an algorithm to solve a new problem with playing cards
 - ▶ Express an algorithm in the Human Machine Language
 - ▶ Identify Sequencing, Selection, and Iteration in a program written in the Human Machine Language
 - ▶ Describe the properties of the Human Machine Language that make it a “low level” language
- 

Two Points from yesterday:

- ▶ Different algorithms can be developed to solve the same problem
 - ▶ Different programs (or code) can be written to implement the same algorithm
- 

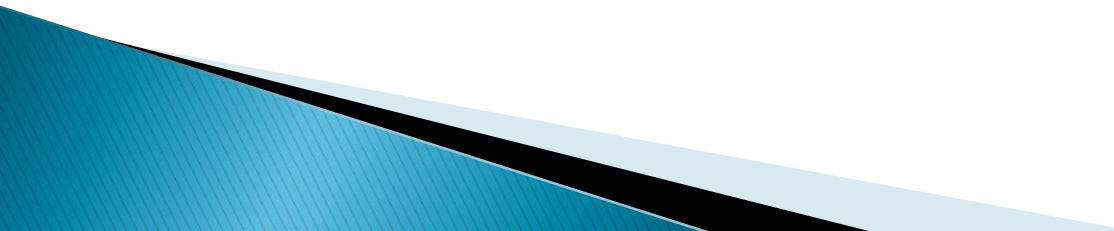
Selection:

- ▶ Also known as “branching” most commonly seen in if–statements is a form of selection.
- ▶ It gives us a way to compare two things (numbers) and take action if one thing was true

Iteration:

- ▶ Also known as “looping”
- ▶ the JUMP command in the Human Machine Language allows us to move to a different point in the program and start executing from there. This allows us to re-use lines of code, and this is a form of iteration or looping

Sequencing:

- ▶ “Sequencing is the application of each step of an algorithm in the order in which the statements are given” (4.4.1 B)
 - ▶ Sequencing is so fundamental to programming it sometimes goes without saying.
 - ▶ In our lesson, the sequencing is simply implied by the fact that we number the instructions with the intent to execute them in order
- 

Sequencing example:

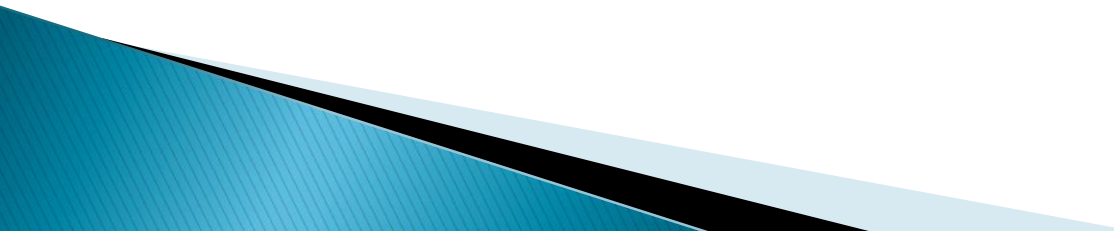
- ▶ $x = 2$
- ▶ $x = 5$
- ▶ $x = x + 1$

- ▶ What is x at the end?

Getting Started: Creativity in Algorithms

- ▶ One thing about algorithms is that once you know a few, and know how they work, you can combine them (or slightly modify them) to solve new problems.
 - Example: find a max card...
- ▶ Creativity in algorithms comes from figuring out clever ways to solve problems by developing a process that could be executed by a machine

Activity: Adding SWAP to Human Machine Language

- ▶ We are going to use the “Human Machine Language – Part 2: Min to Front – Activity Guide”
 - ▶ We will work in pairs
 - ▶ Now: Do the example program
- 

What does the example program do?

- ▶ END STATE: the order of the cards has been reversed
- ▶ It does this by first moving the right hand to the end of the list, then shifting each hand progressively toward the middle of the row, swapping cards each time
- ▶ The program stops once the hands have crossed over each other (by checking if $RHPos < LHPos$)

Challenge: Min-to-Front

- ▶ The challenge is to find the min card and swap it to the front of the list, keeping the order of the rest of the cards the same.
- ▶ Remember: “Algorithms can be combined to make new algorithms” (4.1.1E)

Extra Challenges:

- ▶ Do these if you finished the min to front activity

Wrap-up

- ▶ Trade algorithms with the group next to you (to test the solutions)

Identify Sequencing, Selection, Iteration in Human Machine programs:

- ▶ The CSP Framework states:
 - 4.1.1A Sequencing, selection, and iteration are building blocks of algorithms
 - 4.1.2G Every algorithm can be constructed using only sequencing, selection, and iteration
- ▶ Let's see if we can find this in our programs

Sequencing

- ▶ “4.1.1B Sequencing is the application of each step of an algorithm in the order in which the statements are given.” – Does our human machine language have sequencing?

Selection:

- ▶ “4.1.1C Selection uses a [true–false] condition to determine which of two parts of an algorithm is used.” – Where did we see “selection” in our human machine language programs?

Iteration

- ▶ “4.1.1D Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times.” – Where did we see iteration in our human machine language programs?

Discussion:

- ▶ Algorithms can be combined to make new algorithms
 - Did anyone do this in there program today?

Discussion:

- ▶ Low-level languages exist:
 - Most programming languages that you use in every day life are simply higher level, perhaps easier-to-read commands that are translated into more primitive machine commands
 - So-called “low level” languages are the most basic, primitive, set of commands to control a computer.
 - The Human Machine Language is similar to something called Assembly Language
 - Because the commands are primitive and tie directly to specific functions of the “human machine”
- ▶ “Code in a programming language is often translated into code in another (lower level) language to be executed on a computer” (2.2.3C)

Video

- ▶ You Should Learn to Program: Christian Genco at TEDxSMU
- ▶ Learning to program is really learning how to think in terms of algorithms and processes