Day 2:

* What is pseudocode?
  + What is code?
  + What is a programming language?
    - A way for humans an computers to interact
    - Must bridge the gap between human language (English) and computer language (binary)
* Variables
  + We use variables to remember things
  + We use variables to tell stories
  + Tell a story using variables:
    - Variables.txt pseudocode
    - Focus on reading variable assignments from right to left
    - Focus on using variables for computations
    - Focus on change values of variables
    - Walk through the code an track the value of each variable
  + Ask them to tell a story using variables (10 minutes to work on this)
    - One variable per person
    - Be careful with order of operations!
    - Walk through the solution step by step
* Choice
  + Computers are intelligent enough to make decisions
  + We must provide them with enough information to make these decisions
  + Example: MaxWithEquals.txt
    - Three possible outcomes (x > y, x < y, x == y)
    - Our code can decide what to do based on the outcome
  + Example: MaxNested.txt
    - Sometimes decisions depend on other decisions
    - This leads to nested ifs
    - Walk through an example that uses nested/unnested decisions
  + Exercise: swap (10 minutes to work on this)
    - Requires an if statement
    - Does it require an else?
    - Swap can be tricky – need a variable! Walk through it step by step to show why variable is needed
* Iteration
  + Sometimes we need to repeat things
  + I can ask you for 100 values – do you want to create 100 variables?
  + Instead, we could maybe create one variable and reuse it 100 times
  + Explain the pieces of a loop
    - Initialization
    - Condition
    - Increment
  + Example: Loops.txt
    - Show each required piece of a loop
    - May want to focus on only for loops if time is running short
  + Walk through forLoopExample.txt
    - Show execution of this loop step by step
    - Highlight the three pieces of the loop
  + Exercise: loop writing (5 minutes)
    - These problems all involve changing the pieces of a loop definition
  + Exercise: Fibonacci (10 minutes)
    - Must define necessary variables, before the loop!
    - Must set variable values up for next iteration
  + Exercise: Fizzbuzz (10 minutes)
    - If statements and loops can be used together!
* Functions
  + This concept is taken directly from the mathematical definition of a function
    - A placeholder is used for the input
    - We provide the input value
    - The function substitutes that value in for the placeholder
    - The function gives us a result
  + Example: mathFunction.txt
    - Highlight pieces required for a function definition
      * Output type
      * Name
      * Input type
    - This is the same example as shown on the slide
    - Nice example of a mathematical function
  + Not all functions will be “mathematical”
    - Sometimes functions perform tasks other than math
    - Are inputs necessary?
    - Are outputs necessary?
    - Functions are a way of abstracting a task
    - Instead of repeating the steps of a task, write them down once then refer back to the steps as necessary
  + Example: maxFunction.txt
    - The stuff we have discussed so far (loops, ifs, variables) are fair game for functions!
  + Exercise: Fibonacci function (5 minutes)
    - Turn Fibonacci into a function
    - Code can remain mostly the same, just need to add the function definition!
  + Exercise: Mult (10 minutes)
    - A bit more difficult
    - Uses a loop to perform the multiplication with repeated addition