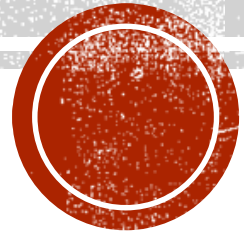
UNIT 3 LESSON 3

Creativity in Algorithims



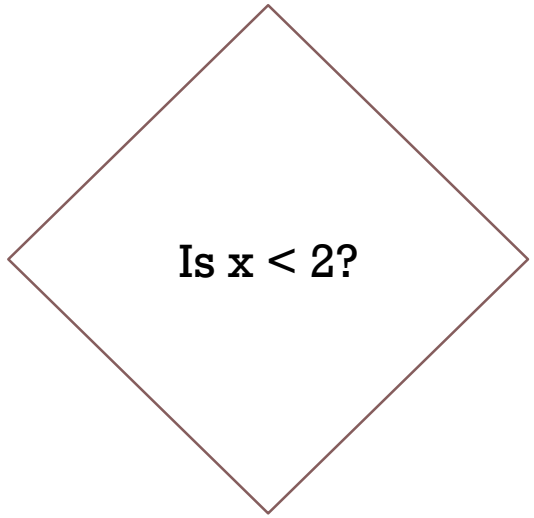
OVERVIEW

- The purpose of this lesson is to see what "creativity in algorithms" means. Creativity has
- to do with both the process you invent (an algorithm) to solve a new problem in a given
- context AND how you implement that algorithm in a given language. Creativity often
- means combining or using algorithms you know as part of a solution to a new problem.
- Thus, the "Min To Front" problem is interesting because students already solved part of it
- (the find min part) in the previous lesson.

- Remember from the last lesson:
- 1. Different algorithms can be developed to solve the same problem
- 2.Different programs (or code) can be written to implement the same algorithm.

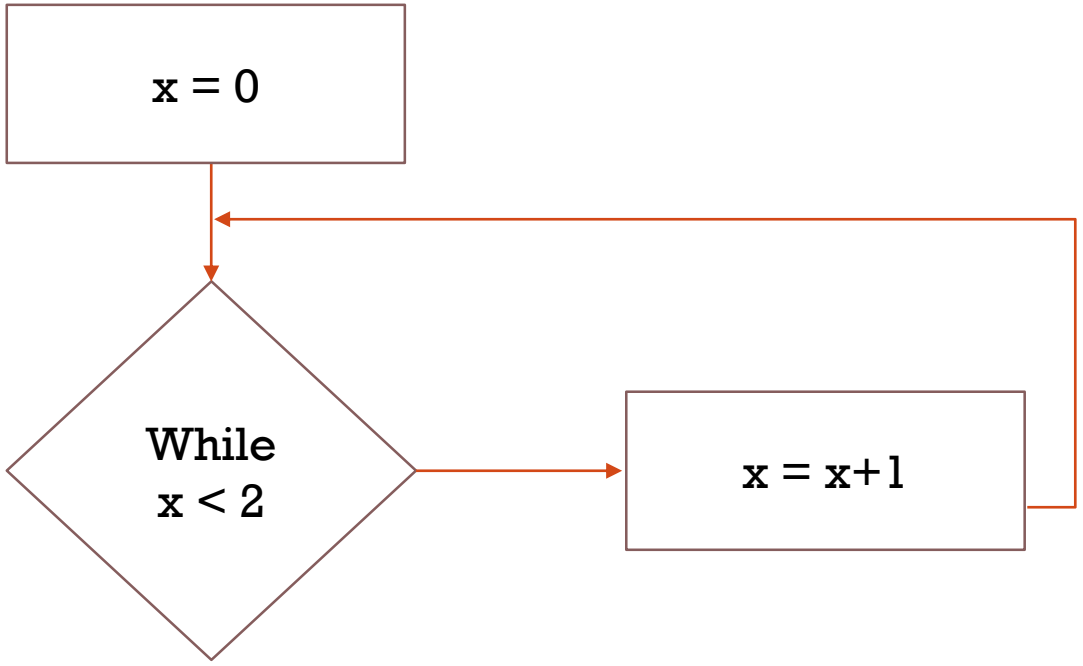
SELECTION/CONDITION/IF-THEN

- **SELECTION**: also known as "branching" most commonly seen in if statements
- – The JUMP…IF command in the Human Machine Language is a form of selection. It gives
- us a way to compare two things (numbers) and take action IF one thing was true.



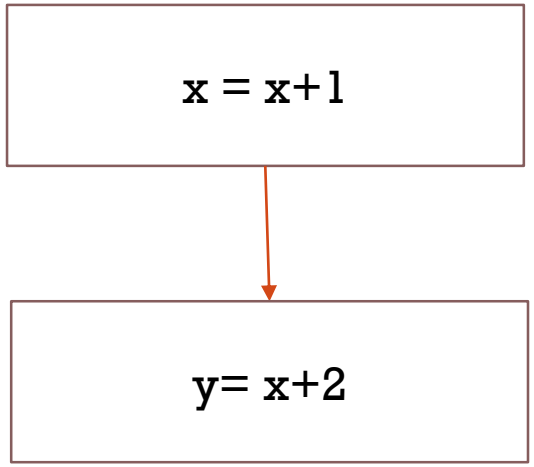
ITERATION/LOOP

- **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 reuse lines of code, and this is a form of iteration or
- looping.



SEQUENCING

- **SEQUENCING:** From the framework: "4.1.1B Sequencing is the application of each
- step of an algorithm in the order in which the statements are given." 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.



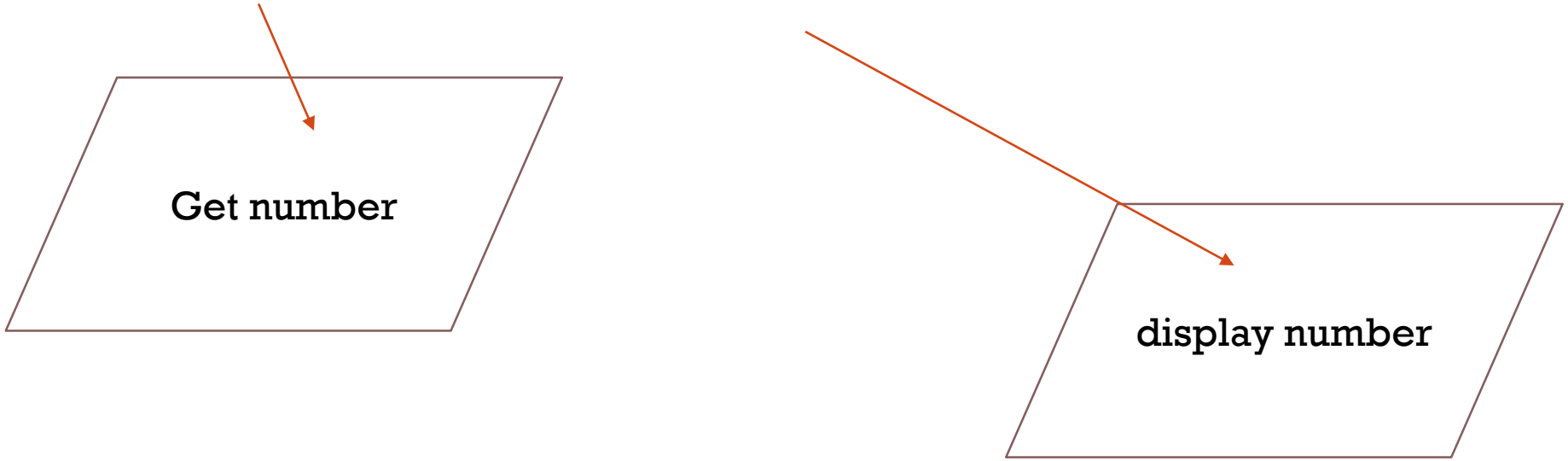
VARIABLE DECLARATION

- For I/O & Processing we usually need variables or a place to store an unknown value that the user enters OR that we calculate.
- For now, use camel casing to name them and they are of type:
- String(Words), Num, or Boolean(T/F)

num price
string name
boolean weekDay

INPUT/OUTPUT

- When input is entered OR output is displayed in code this symbol is used



TERMINATOR

- When code begins there is a symbol that represents the start of the code. Likewise, when the code ends or terminates, that same symbol is used. It is normally only used once to start the code and once to end the code.

FLOW CHARTING/PSEUDOCODE EXAMPLE

- You want the user to enter a value for two sides of a right triangle. Compute the hypotenuse and print it out.
- More Real World: Draw a flowchart and write pseudo code (on the same sheet) to represent the logic of a program that allows the user to enter three values. The first value represents an hourly pay rate, the second represents the number of hours worked this pay period, and the third represents the percentage of gross salary withheld. The program multiplies the hourly pay rate by the number of hours worked, giving the gross pay; then, it multiplies the gross pay by the withholding percentage, giving the withholding amount. Finally, it subtracts the withholding amount from the gross pay, giving it net pay after taxes. The program prints the net pay.

FLOW CHARTING/PSEUDOCODE EXAMPLE

- More Real World: Rick Hammer is a carpenter who wants an application to compute the price of any desk a customer orders, based on the following: desk length and width in inches, type of wood, and number of drawers. The price is computed as follows:
- The minimum charge for all desks is $200.
- If the surface (length * width) is over 750 square inches, add $50.
- If the wood is "mahogany" add $150; for "oak" add $125. No charge is added for "pine."
- For every drawer in the desk, there is an additional $30 charge.

- Design a flowchart and pseudocode for a program that accepts data for an order number, customer name, length and width of the desk ordered, type of wood, and number of drawers. Display all the entered data and the final price for the desk.

COMMON ALGORITHMS TO KNOW IN CS

- Nearest Neighbor (APCSP)
- Linear versus Binary Search (APCSP)
- Sorting Data (APCSA/Kahn Academy):
  - Selection Sort
  - Insertion Sort
  - Merge Sort
  - Quick Sort
  - Bubble Sort
- Recursive Algorithms (APCSA/Kahn Academy)
- Towers of Hanoi (APCSA/Kahn Academy)
- Graph representation (Kahn Academy)
- Breadth-First Search (Kahn Academy)

SUMMARY & HOMEWORK

- HOMEWORK: Complete U3L3 Reflection PINK SHEET
  - due at the start of class tomorrow