Announcements

• Assignment 6 is due today!
• Today
  – Chap 10 – variables and arrays
Review: Properties

- Where is the class?
- Where is the object?
- A class defines properties
- When an object is created it receives its own set of properties
State and Changing State

- State of object – each property stores info about the object
  - Example:
    - vehicle
    - isShowing

- State change
Class-level Variables

- New variables can be added to the properties of an object – class-level
  - The value of the variable can be changed
    - The variable is **mutable**.
    - Can be used to track state changes.
Inheritance

• If an object (and its new variable) are saved out and given a new name, a new class is created.
  – This is inheritance!
  – The new class **inherits** all properties and methods of original class.
    • Did this before by adding new methods to a class and saving it out.
What is an array?

• An array is a collection of objects or information organized in a specific order.

• The individual components (elements) are of the same type (all object or all number, etc.)

• Analogy – Music CD
  – Collection of songs listed in order
  – CD player allows you to
    • Play songs in order
    • Play songs by specifying its number
    • Play songs in random order
Arrays in Alice

• In Alice, array is a data structure to organize objects or information
• An array is not visible, it is a way of organizing
• But….
  – Alice has a 3D model to help you “see” the array
Example – Create a visualization of an array of people

• Add 5 people to the world
• Add an array visualization
• Not an array yet, must add people to the array
• Positions in array numbered starting with 0
Initialize array - Add Alice to Array in position 0

- Alice automatically moves to the 0 position!
Add Soldier to the Array

- Soldier moves automatically to position 1 (which is the 2cd position)!
Add RandomGuy, Skater and Rockette

• The array initialization is complete!

• Set isVisible for arrayVisualization to false
  – Array not seen
Accessing elements in an array

- Can specify an element at a particular location in the array
Repeat for all items in the array – in order
• Use “loop” – complicated version
• Note: index in loop is used in body
• What does this do?

• What does this do?
Array vs. List

• Array
  – Elements are ordered
  – Can access a particular element – 3
  – Use “Loop” - loop over elements – one at a time, OR every second element, etc

• List
  – Elements are not ordered
  – Use “For all in order”, “For all together” – does something to each element in the list – just don’t know the order this occurs
Swapping two elements in the array

• Swap the objects at positions 0 (fanDancer) and 3 (duckPrince)
• Add in an ObjectVisualization, this is like a variable for an object. (same folder where ArrayVisualization is)
Swapping objects at 0 and 3 (cont)

• Only one element at a time can be in a slot in the array. To swap two elements, you have to move one of them out temporarily.

• Move object at index 0 to objectVisualization (this frees up slot 0)
Swapping objects at 0 and 3 (cont)

- Now you can move the item in slot 3 over to slot 0 (note the duckPrince moved over)
- Now slot 3 is empty
Swapping objects at 0 and 3 (cont)

• Now move the object that was originally in slot 0 and was saved temporarily in the ObjectVisualization, over to slot 3
Setting elements in array

- Objects in an array are called **elements**
- Use “let” to set a position in an array
- Using “let”:

```
let ArrayVisualization[0] = item
```

Don’t do this if there is already something in position 0! Move the item first!
Swapping objects at 0 and 3 (code)

• Here is the code that corresponds to the swapping of the items in slots 0 and 3.
Shuffle the array

- For each item in the array, swap it randomly with another object
SelectionSort the array

- Find the position of the shortest object
  - Swap that object with the object in position 0
- Find the position of the next shortest object
  - Swap that object with the object in position 1
- Etc…. Until the array is sorted.
Classwork Today

• Shuffle Array
• Sort Array