Note: Much of this material will be covered on Wednesday

- Academistrative
  - Where to find coding examples
  - Midterm on Friday
    - In classroom
  - HW due the following Friday
- Last time
  - Generics
  - Began to understand how to create a purely-object oriented array using generics ("Array<T>")
  - Hinted at how we can extend this array to build some kind of magic array that expands at will
- Today
  - Continue our work from last time, moving towards the magic array
  - Formally introduce the List abstract data type from the book
  - Talk about its relationship with List interface in Java
  - Begin to implement ArrayList
- Purely object-oriented array
  - FixedSizeArray.java
  - Demo some of the cool things we can do
- Expandable array
  - First let's make an Array interface as there are a few things any array should be able to do
  - Array.java
  - Expanded
- Introduce List ADT
  - Talk about its relationship with List interface in Java
    - Go to list interface
    - Talk about relationship between our ExpandableArray
- Begin to implement ArrayList
  - ArrayList does this exactly
  - ArrayList is your new best friend
    - Very straightforward
    - No magic
    - Expandable array
  - We'll begin to walk through formal implementations on Wednesday

- Why List interface?
  - There are other types of lists with computational advantages in certain cases!
- Previews and Miscellany
  - **Comparable interface**
    - **Collections.sort();**
    - **Implement for RockStar**
  - **Understanding comparisons**
    - **Preview of LinkedList and ArrayList**
      - **Can talk about Linkedlists avoid the need to resize!**
  - Quick point wrt inheritance and generics
    - An ArrayList<String> is not Array<Object>
  - `ArrayList<? extends Object> = ArrayList<?>`
    - https://docs.oracle.com/javase/tutorial/extra/generics/subtype.html
    - Actual parent class of all ArrayList<?>
  - Counting # of comparisons