Today’s topics

• Topics
  • Class Scores revisited
  • ArrayLists & Sets
  • Counting unique words
• **Library.** A module whose methods are primarily intended for use by many other programs.

• **Client.** Program that calls a library.

• **API.** Contract between client and implementation.

• **Implementation.** Program that implements the methods in an API.
• An array that does not have fixed length

ArrayList<String> list = new ArrayList<String>();

// add to ArrayList
list.add("hello");

//check if element is in ArrayList
boolean inList = list.contains("hello");

//get element at index five
String word = list.get(5);
• No primitives! Only Objects!

```java
ArrayList<int> list = new ArrayList<int>();

ArrayList<Integer> list = new ArrayList<Integer>();
```
Problems

1. How do you print all of the elements of an array of ints?
   • Add a method `printArray` that will print all the elements one per line
   • What if you wanted to print in standard form?
     \[ [88, 70, 65, 70, 88] \]

2. How would `printArrayList` differ?

3. How would you copy from an array to an `ArrayList` and vice versa?
Sets

- Set is an unordered list of items
  - Items are unique! Only one copy of each item in set!
- We will use two different implementations of sets
  - `java.util.TreeSet`
    - A TreeSet is backed up by a tree structure (future topic)
    - Keeps items sorted (+)
    - Slower than HashSets ?? (-)
  - `java.util.HashSet`
    - A HashSet is backed up by a hashing scheme (future topic)
    - Items not sorted – should seem to be in random order (-)
    - Faster than TreeSets ?? (+)
0. “Hey, I have some data that are too complicated for a primitive (or array of primitives).”
1. Decide what data you need. (nouns)
2. Decide what methods you need (verbs)
3. Make a new class:
4. Add your data
   Inside the class;
   outside the methods
5. Write your constructor
   (which fills in your data)
6. Write the rest.
Problem

• How many different/unique words are in a file?

• How many words do two files have in common?

• Snarf *classwork/03_WordCount2*
  • Submit as *class-0903*