2/1

Collections

- object that stores data; aka "data structure"
  - ArrayList, LinkedList, HashMap, TreeSet, PriorityQueue

    Lists

- a collection storing an ordered sequence of elements
  - each element is accessible by a 0-based index
  - has a size
  - elements can be added to front, back, or elsewhere
  - can be represented as an ArrayList object

    Idea of a List

- create an object that represents a "list" of items
- can add idtems to the list
- list object keeps track of the element values that have been added to it, their order, indexes, and its total size
  - automatically resizing array object
  - implemented using an array and a size field

    Type Parameters (Generics)

    `ArrayList<Type> name = new ArrayList<Type>();`

- when constructing an ArrayList, must specift the type of elements it will contain between < and >

  ```java
  ArrayList<String> names = new ArrayList<String>();
  names.add("Marty Stepp");
  names.add("Stuart Reges");
  ```

  ArrayList vs. Array

- Construction `String[] names = new String[5];` `ArrayList<String> list = new ArrayList<String>();`
- Storing a value `names[0] = "Jessica";` `list.add("Jessica");`
- Retrieving a value `String s = names[0];` `String s = list.get(0);`
- Doing something to each value that starts with "B"

  ```java
  for(int i = 0; i < names.length; i++) {
    if(names[i].startsWith("B") { ... }
  }
  for(int i = 0; i  < list.size(); i++) {
    if(list.get(i).startsWith("B") { ... }
  }
  ```

- Seeing whether the value "Benson" is found

  ```java
  for(int i = 0; i < names.length; i++) {
    if(names[i].equals("Benson")) { ... }
  }
  if(list.contains("Benson")) { ... }
  ```

Exercise

- Write a program that reads a file and manipulates the words of that file as a list
    - Read all words from file
    - Reverse the order of the words
    - Capitalize all plurals (ending in "s")
    - Remove all plural words

```
import java.io.File;
import java.util.ArrayList;
import java.util.Scanner;
```

public class Main { public static ArrayList readWordsFromFile(String file) { ArrayList words= new ArrayList(); try { Scanner s = new Scanner(new File(file)); while(s.hasNext()) { String word = s.nextLine(); words.add(word); return words; } } catch (FileNotFoundException ex) { System.out;println("File not found"); return null; } }

```
public static ArrayList<String> reverse(ArrayList<String> words) {

}

public static ArrayList<String> capitalize(ArrayList<String> words) {

}

public static ArrayList<String>
```

}

public class MainTest { @Test public void testReadWordsFromFile() { ArrayList actual = Main.readWordsFromFile("words.txt"); ArrayList expected = new ArrayList(); expected.add("foo"); expected.add("bar"); expected.add("baz"); expected.add("test"); assertEquals(expected, actual); } }

words.txt foo bar baz test

~~~
#### Wrapper Classes
- an object whose sole purpose is to hold a primitive value
- Once list is constructed, use it with primitives as normal
```java
ArrayList<Double> grades = new ArrayList<Double>;
grades.add(3.2);
grades.add(2.7);
...
double myGrade = grades.get(0);
~~~