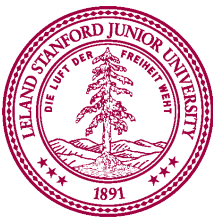
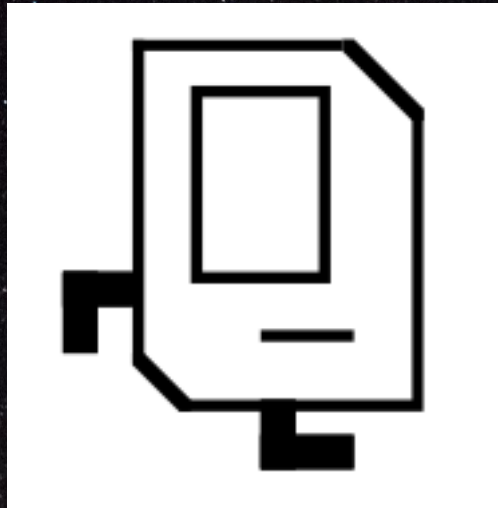


<https://xkcd.com/347/>





File Processing & ArrayLists

Nick Troccoli

CS106A, Stanford University

Learning Goals

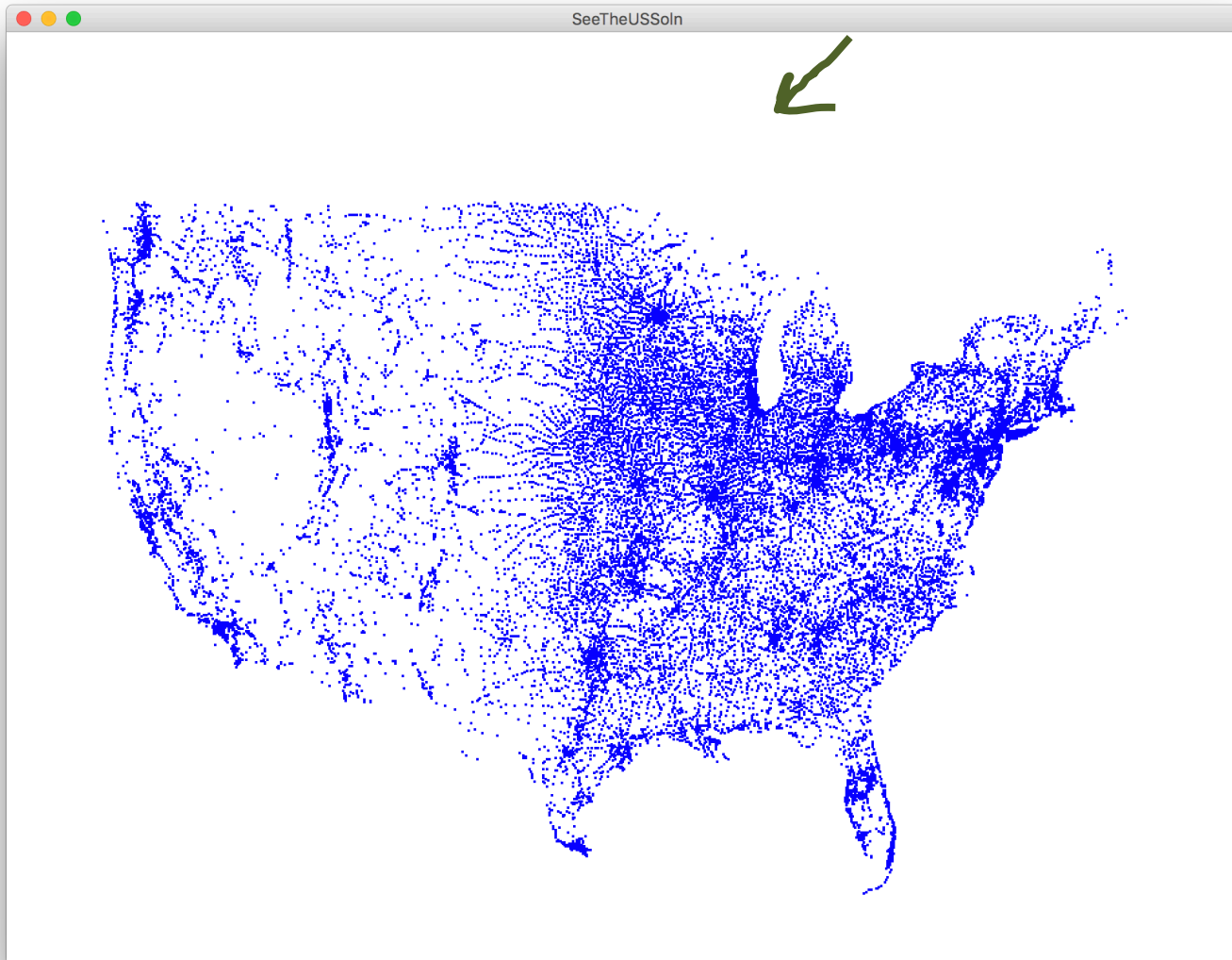
1. Know how to read a file line by line.
2. Know how to store and retrieve data from an ArrayList.



File Processing

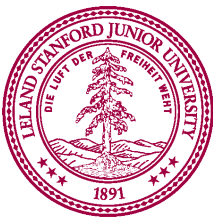
Thanks Keith Schwarz for some great slides to build off!

Each blue dot is a tiny
GRect



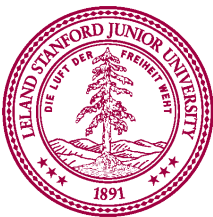
Files Are Cool!

- Virtually all programs that you've used at some point read files from disk:
 - Word processing (documents)
 - Web browser (cookies)
 - Games (saved progress)
 - Eclipse (Java files)
 - Music player (songs)



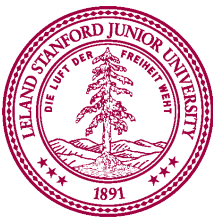
The structure of files

- A file is just a series of *bits* (ones and zeros).
- Those bits can have structure:
 - Plain-text: Bits represent characters.
 - JPEG: Bits encode information about the structure of an image.
 - MP3: Bits encode frequency information about music.
 - etc.

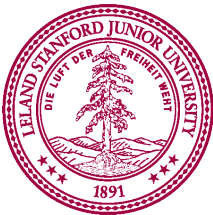


The structure of files

- A file is just a series of *bits* (ones and zeros).
- Those bits can have structure:
 - **Plain-text:** Bits represent characters.
 - JPEG: Bits encode information about the structure of an image.
 - MP3: Bits encode frequency information about music.
 - etc.



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"



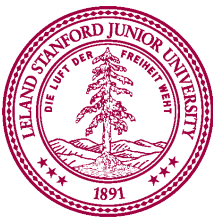
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

Step one:
Open the file for reading.



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));
```

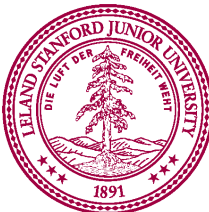


Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));
```

To use the `BufferedReader` and `FileReader`
types, you need to

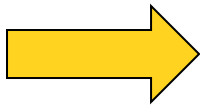
```
import java.io.*;
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));
```

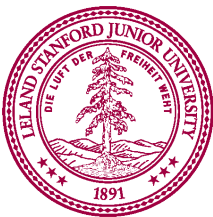


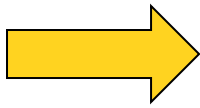


Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));
```

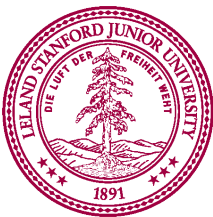
Step Two:
Read the file,
one line at a time.

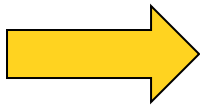




Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine();
```





Yesterday, upon the stair,

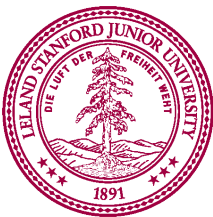
I met a man who wasn't there

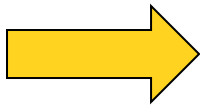
He wasn't there again today

I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine();
```





Yesterday, upon the stair,

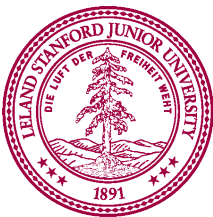
I met a man who wasn't there


He wasn't there again today

I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

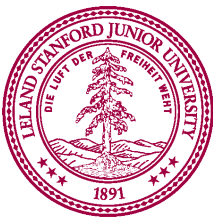
```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,
```






Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

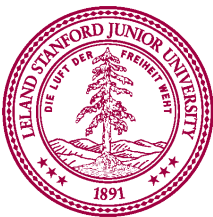
```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,
```





Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine();
```





Yesterday, upon the stair,

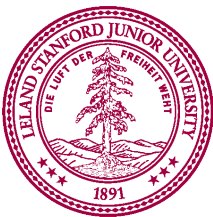
I met a man who wasn't there

He wasn't there again today

I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine();
```





Yesterday, upon the stair,

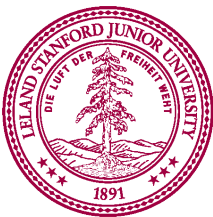
I met a man who wasn't there

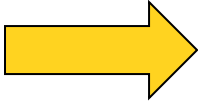
He wasn't there again today

I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

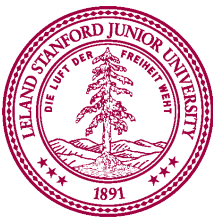
```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there
```

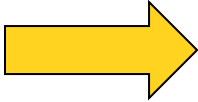




Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

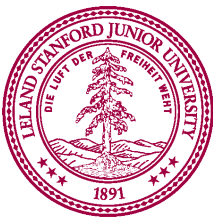
```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there
```





Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine();
```



Yesterday, upon the stair,

I met a man who wasn't there

He wasn't there again today

I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

BufferedReader br =

```
new BufferedReader(new FileReader("poem.txt"));
```

```
String line1 = br.readLine(); // Yesterday, upon the stair,
```

```
String line2 = br.readLine(); // I met a man who wasn't there
```

```
String line3 = br.readLine();
```



Yesterday, upon the stair,

I met a man who wasn't there

He wasn't there again today

I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

BufferedReader br =

```
new BufferedReader(new FileReader("poem.txt"));
```

```
String line1 = br.readLine(); // Yesterday, upon the stair,
```

```
String line2 = br.readLine(); // I met a man who wasn't there
```

```
String line3 = br.readLine(); // He wasn't there again today
```

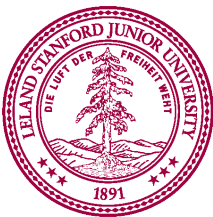



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

BufferedReader br =

```
new BufferedReader(new FileReader("poem.txt"));
```

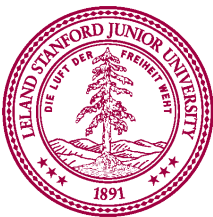
```
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today
```





Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine();
```



Yesterday, upon the stair,

I met a man who wasn't there

He wasn't there again today

I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"



```
BufferedReader br =
```

```
    new BufferedReader(new FileReader("poem.txt"));
```

```
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine();
```



Yesterday, upon the stair,

I met a man who wasn't there

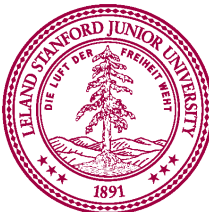
He wasn't there again today

I wish, I wish he'd go away...

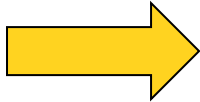
- Hughes Mearns, "Antagonish"



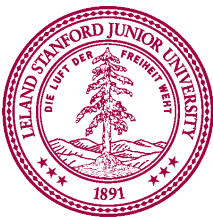
```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...
```



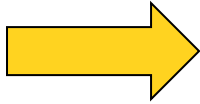
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"



```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"



```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine();
```



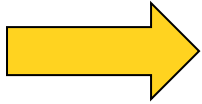
Yesterday, upon the stair,

I met a man who wasn't there

He wasn't there again today

I wish, I wish he'd go away...

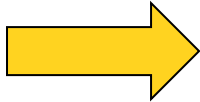
- Hughes Mearns, "Antagonish"



```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine();
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

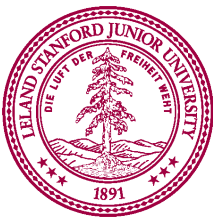


```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
String line6 = br.readLine();
```



Yesterday, upon the stair,

I met a man who wasn't there

He wasn't there again today

I wish, I wish he'd go away...

- Hughes Mearns, "Antagonish"

```
BufferedReader br =
```

```
    new BufferedReader(new FileReader("poem.txt"));
```

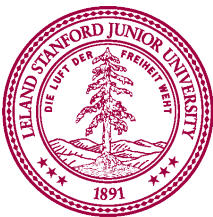
```
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
String line6 = br.readLine(); // *Returns null*
```



Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
String line6 = br.readLine(); // *Returns null*
```

Step Three:
Close the file.



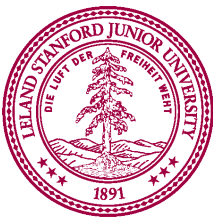

```
Yesterday, upon the stair,  
I met a man who wasn't there  
He wasn't there again today  
I wish, I wish he'd go away...  
- Hughes Mearns, "Antagonish"
```

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line1 = br.readLine(); // Yesterday, upon the stair,  
String line2 = br.readLine(); // I met a man who wasn't there  
String line3 = br.readLine(); // He wasn't there again today  
String line4 = br.readLine(); // I wish, I wish he'd go away...  
String line5 = br.readLine(); // - Hughes Mearns, "Antagonish"  
String line6 = br.readLine(); // *Returns null*  
  
br.close();
```



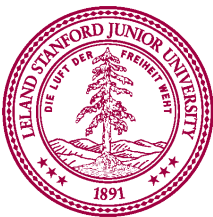
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line = br.readLine();  
while (line != null) { // while there are more lines..  
    println(line); // print out each line  
    line = br.readLine();  
}  
br.close();
```



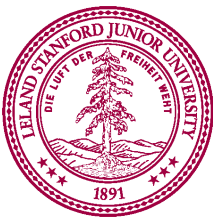
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line = br.readLine();  
while (line != null) {           // while there are more lines..  
    println(line);              // print out each line  
    line = br.readLine();  
}  
br.close();
```



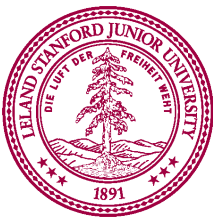
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line = br.readLine();  
while (line != null) { // while there are more lines..  
    println(line); // print out each line  
    line = br.readLine();  
}  
br.close();
```



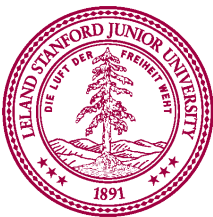
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line = br.readLine();  
while (line != null) { // while there are more lines..  
    println(line); // print out each line  
    line = br.readLine();  
}  
br.close();
```



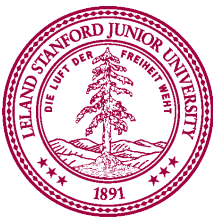
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line = br.readLine();  
while (line != null) {                // while there are more lines..  
    println(line);                    // print out each line  
    line = br.readLine();  
}  
br.close();
```



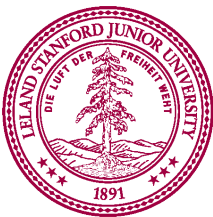
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line = br.readLine();  
while (line != null) {           // while there are more lines..  
    println(line);              // print out each line  
    line = br.readLine();  
}  
br.close();
```



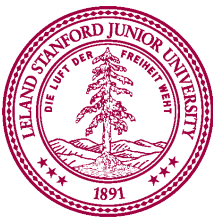
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line = br.readLine();  
while (line != null) {           // while there are more lines..  
    println(line);              // print out each line  
    line = br.readLine();  
}  
br.close();
```



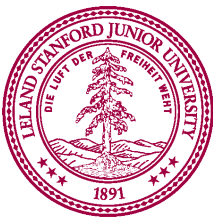
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line = br.readLine();  
while (line != null) {           // while there are more lines..  
    println(line);              // print out each line  
    line = br.readLine();  
}  
br.close();
```



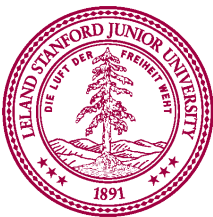
Yesterday, upon the stair,
I met a man who wasn't there
He wasn't there again today
I wish, I wish he'd go away...
- Hughes Mearns, "Antagonish"

```
BufferedReader br =  
    new BufferedReader(new FileReader("poem.txt"));  
  
String line = br.readLine();  
while (line != null) { // while there are more lines..  
    println(line); // print out each line  
    line = br.readLine();  
}  
br.close();
```



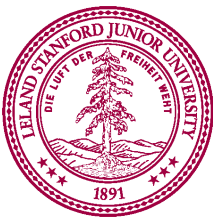
Sometimes things break

- Programs sometimes encounter unexpected errors.
- Sometimes these are bugs:
 - Dividing by zero.
 - Sending a message to a **null** object.
- Sometimes these are due to external factors:
 - Network errors.
 - Missing files.



Exceptional cases

- An **exception** occurs if Java encounters a case where it can't proceed as normal.
- Java requires that your program handle certain types of exceptions.
- Think of exceptions as rerouting control in an emergency:
 - If all goes well, program continues as usual.
 - If something goes wrong, handle the emergency.
- File processing exceptions: file not found, corrupted, etc.

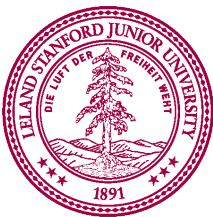


try-ing your best

- To use a method or class that might cause an exception, you must tell Java to **try** its best, knowing it might fail.

```
try {
    BufferedReader br =
        new BufferedReader(new FileReader("poem.txt"));

    String line = br.readLine();
    while (line != null) {           // while there are more lines...
        println(line);             // print out each line
        line = br.readLine();
    }
    br.close();
}
```

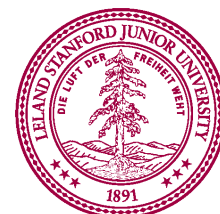


try and catch me

- If an exception occurs, you need to tell Java to **catch** that exception, and what to do after that.

```
try {
    BufferedReader br =
        new BufferedReader(new FileReader("poem.txt"));

    String line = br.readLine();
    while (line != null) {           // while there are more lines...
        println(line);              // print out each line
        line = br.readLine();
    }
    br.close();
} catch (IOException e) {
    // Handle the exception if there is one
    println("An error occurred: " + e);
}
```



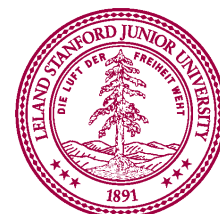
try and catch me

- If an exception occurs, you need to tell Java to **catch** that exception, and what to do after that.

If something fails up here...

```
try {
    BufferedReader br =
        new BufferedReader(new FileReader("poem.txt"));

    String line = br.readLine();
    while (line != null) {           // while there are more lines...
        println(line);              // print out each line
        line = br.readLine();
    }
    br.close();
} catch (IOException e) {
    // Handle the exception if there is one
    println("An error occurred: " + e);
}
```



try and catch me

- If an exception occurs, you need to tell Java to **catch** that exception, and what to do after that.

If something fails up here...

```
try {
    BufferedReader br =
        new BufferedReader(new FileReader("poem.txt"));

    String line = br.readLine();
    while (line != null) {           // while there are more lines...
        println(line);              // print out each line
        line = br.readLine();
    }
    br.close();
} catch (IOException e) {           ... we immediately
    // Handle the exception if there is one
    println("An error occurred: " + e);
    jump down here.
}
```

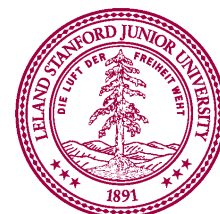


try and catch me

- If an exception occurs, you need to tell Java to **catch** that exception, and what to do after that.

```
try {
    BufferedReader br =
        new BufferedReader(new FileReader("poem.txt"));

    String line = br.readLine();
    while (line != null) {           // while there are more lines...
        println(line);              // print out each line
        line = br.readLine();
    }
    br.close();
} catch (IOException e) {
    // Handle the exception if there is one
    println("An error occurred: " + e);
}
```

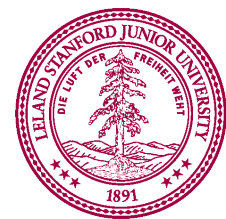


try and catch me

- If an exception occurs, you need to tell Java to **catch** that exception, and what to do after that.

```
try {
    BufferedReader br =
        new BufferedReader(new FileReader("poem.txt"));

    String line = br.readLine();
    while (line != null) {           // while there are more lines...
        println(line);             // print out each line
        line = br.readLine();
    }
    br.close();
} catch (IOException e) {
    // Handle the exception if there is one
    println("An error occurred: " + e);
}
```

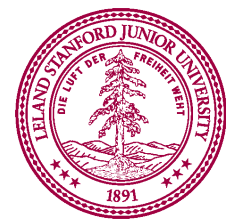


try and catch me

- If an exception occurs, you need to tell Java to **catch** that exception, and what to do after that.

```
try {
    BufferedReader br =
        new BufferedReader(new FileReader("poem.txt"));

    String line = br.readLine();
    while (line != null) {           // while there are more lines...
        println(line);             // print out each line
        line = br.readLine();
    }
    br.close();
} catch (IOException e) {
    // Handle the exception if there is one
    throw new RuntimeException(e);
}
```

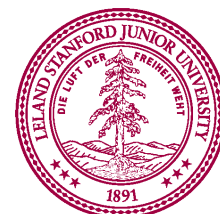


try and catch me

- If an exception occurs, you need to tell Java to **catch** that exception, and what to do after that.

```
try {
    BufferedReader br =
        new BufferedReader(new FileReader("poem.txt"));

    String line = br.readLine();
    while (line != null) {           // while there are more lines...
        println(line);              // print out each line
        line = br.readLine();
    }
    br.close();
} catch (IOException e) {
    // Handle the exception if there is one
    e.printStackTrace();
}
```



File concepts in one slide

1. Make a `BufferedReader` to open a file to read

```
BufferedReader br = new BufferedReader(new FileReader("poem.txt"));
```

2. Use `br.readLine` to get one line from the file, usually in a loop

```
br.readLine(); // returns the next line, or null
```

3. Both the above operations are “dangerous” so we need to use a try/catch block

```
try {  
    // you can do it!  
} catch (Exception e){  
    // ok maybe not  
}
```

4. You can either handle the problem or throw a runtime exception

```
throw new RuntimeException("AHHHH!");
```



Canonical File Processing Program

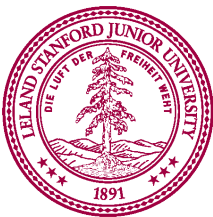
```
try {
    BufferedReader br = /*...open the file... */
    String line = br.readLine();
    while (line != null) {

        /* ... process current line ... */

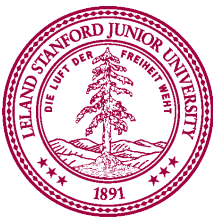
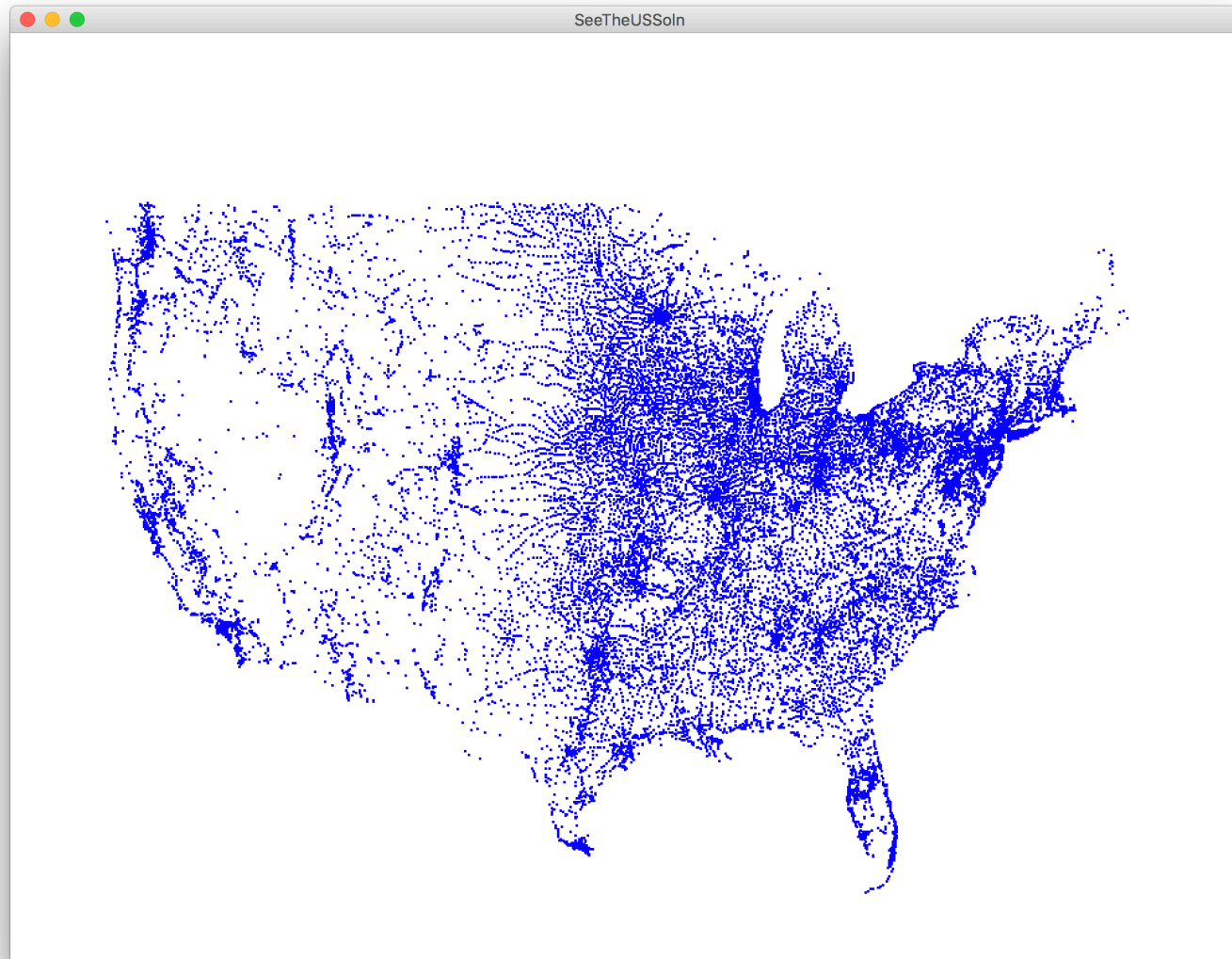
        line = br.readLine(); /* get next line */
    }
    br.close();
} catch (IOException e) {
    throw new RuntimeException(e);
}
```



Understanding this code is about 95%
of what we want you to know for files in
CS106A



Example: US Census Data



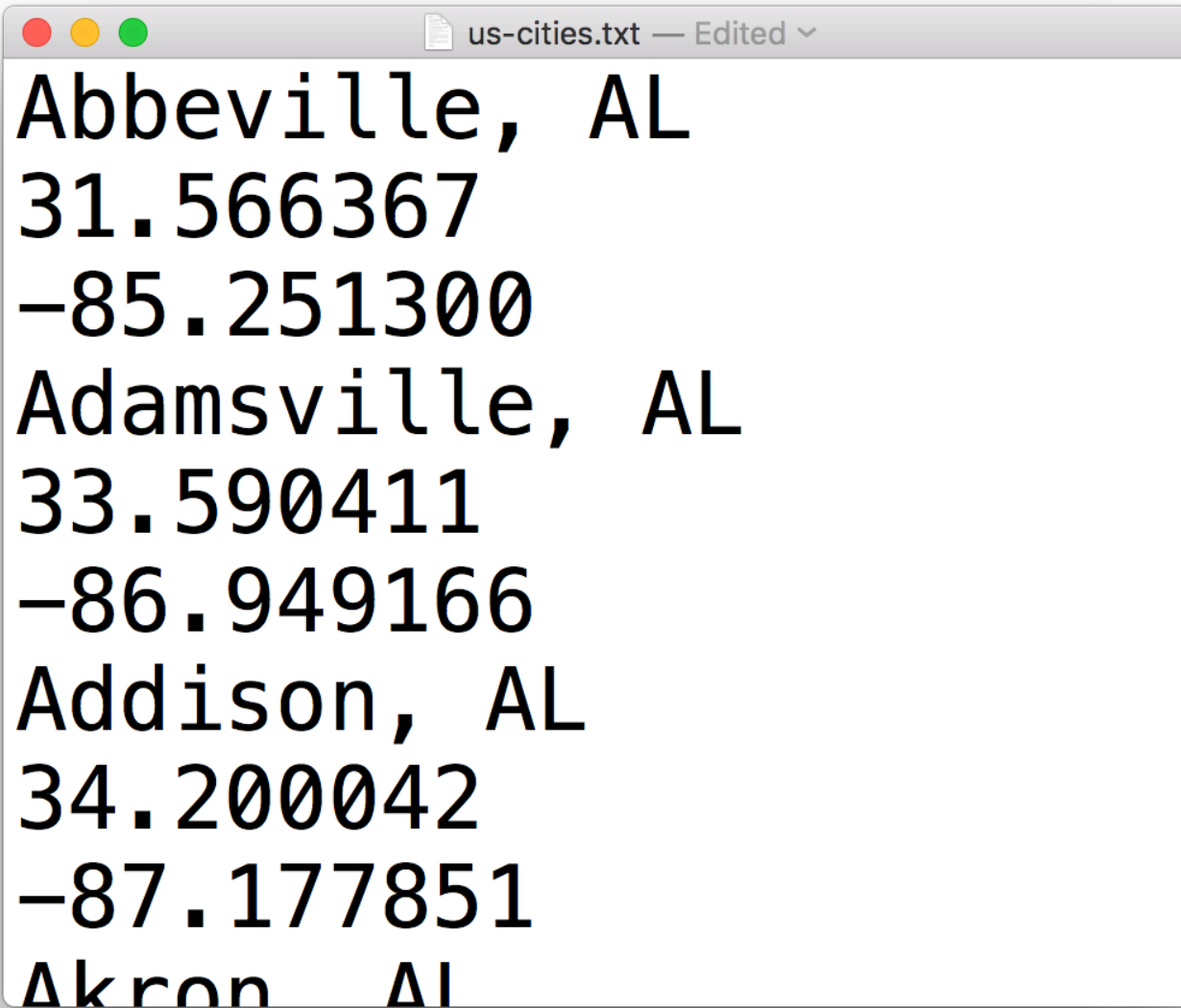
Example: US Census Data

```
us-cities.txt — Edited v
Abbeville, AL
31.566367
-85.251300
Adamsville, AL
33.590411
-86.949166
Addison, AL
34.200042
-87.177851
Akron AL
```

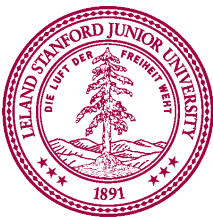


Example: US Census Data

City name



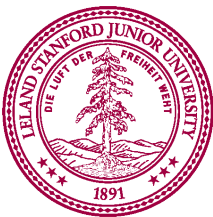
```
us-cities.txt — Edited v
Abbeville, AL
31.566367
-85.251300
Adamsville, AL
33.590411
-86.949166
Addison, AL
34.200042
-87.177851
Akron AL
```



Example: US Census Data

Latitude

```
us-cities.txt — Edited v
Abbeville, AL
31.566367
-85.251300
Adamsville, AL
33.590411
-86.949166
Addison, AL
34.200042
-87.177851
Akron AL
```



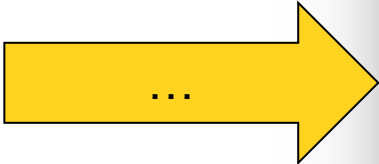
Example: US Census Data



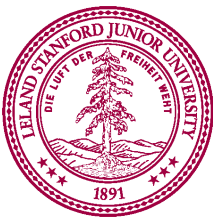
```
us-cities.txt — Edited v
Abbeville, AL
31.566367
-85.251300
Adamsville, AL
33.590411
-86.949166
Addison, AL
34.200042
-87.177851
Akron AL
```



Example: US Census Data



```
us-cities.txt — Edited v
Abbeville, AL
31.566367
-85.251300
Adamsville, AL
33.590411
-86.949166
Addison, AL
34.200042
-87.177851
Akron AL
```



Recap: File Processing

1. Make a `BufferedReader` to open a file to read

```
BufferedReader br = new BufferedReader(new FileReader("poem.txt"));
```

2. Use `br.readLine` to get one line from the file, usually in a loop

```
br.readLine(); // returns the next line, or null
```

3. Both the above operations are “dangerous” so we need to use a try/catch loop

```
try {  
    // you can do it!  
} catch (Exception e){  
    // ok maybe not  
}
```

4. You can either handle the problem or throw a runtime exception

```
throw new RuntimeException("AHHHH!");
```

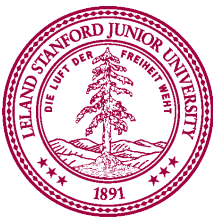


ArrayLists

Thanks Keith Schwarz for some great slides to build off!

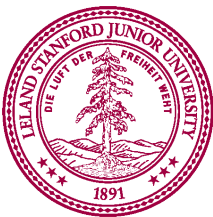
ArrayList

- An ordered, resizable list of information
- Homogeneous
- Can add and remove elements (among other cool functionality)
- Can store any **object** type
- Requires importing **java.util.***;



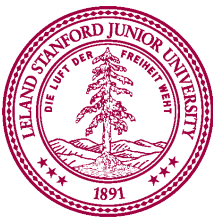
Our First ArrayList

```
ArrayList<String> myArrayList = new ArrayList<String>();
```



Our First ArrayList

```
ArrayList<String> myArrayList = new ArrayList<String>();
```

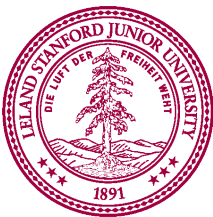


Our First ArrayList

Type of thing your
ArrayList will store.

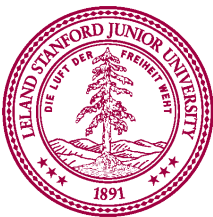


```
ArrayList<String> myArrayList = new ArrayList<String>();
```



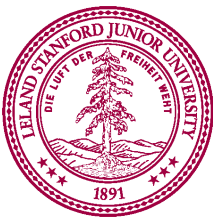
Our First ArrayList

```
ArrayList<String> myArrayList = new ArrayList<String>();
```



Our First ArrayList

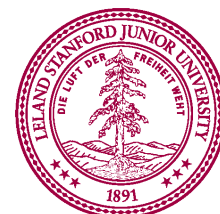
```
ArrayList<String> myArrayList = new ArrayList<String>();
```



Our First ArrayList

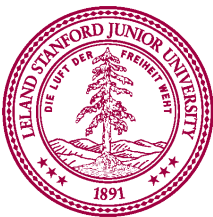
Same type here, but
followed by ().

```
ArrayList<String> myArrayList = new ArrayList<String>();
```



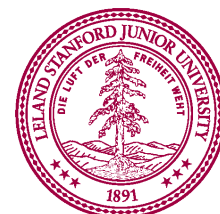
Our First ArrayList

```
ArrayList<String> myArrayList = new ArrayList<String>();
```



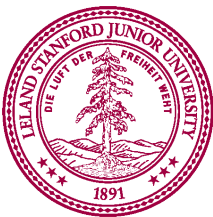
Our First ArrayList

```
ArrayList<String> myArrayList = new ArrayList<String>();  
  
// Adds elements to the back  
myArrayList.add("hi");
```



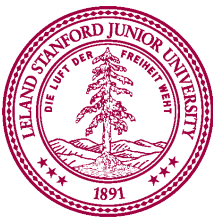
Our First ArrayList

```
ArrayList<String> myArrayList = new ArrayList<String>();  
  
// Adds elements to the back  
myArrayList.add("hi");  
myArrayList.add("there");
```



Our First ArrayList

```
ArrayList<String> myArrayList = new ArrayList<String>();  
  
// Adds elements to the back  
myArrayList.add("hi");  
myArrayList.add("there");  
  
// Access elements by index (starting at 0!)  
println(myArrayList.get(0)); // prints "hi"  
println(myArrayList.get(1)); // prints "there"
```



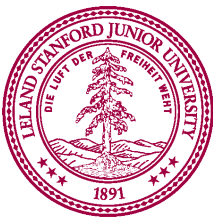
Our First ArrayList

```
ArrayList<String> myArrayList = new ArrayList<String>();

// Adds elements to the back
myArrayList.add("hi");
myArrayList.add("there");

// Access elements by index (starting at 0!)
println(myArrayList.get(0)); // prints "hi"
println(myArrayList.get(1)); // prints "there"

// Wrong type - bad times! Won't compile
GLabel label = new GLabel("hi there");
myArrayList.add(label);
```



Our First ArrayList

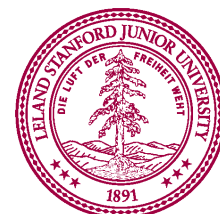
```
ArrayList<String> myArrayList = new ArrayList<String>();

// Adds elements to the back
myArrayList.add("hi");
myArrayList.add("there");

// Access elements by index (starting at 0!)
println(myArrayList.get(0)); // prints "hi"
println(myArrayList.get(1)); // prints "there"

// Wrong type - bad times! Won't compile
GLabel label = new GLabel("hi there");
myArrayList.add(label);

// Invalid index - crashes! IndexOutOfBoundsException
println(myArrayList.get(2));
```



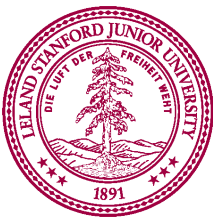
Our First ArrayList

```
ArrayList<String> myArrayList = new ArrayList<String>();

// Adds elements to the back
myArrayList.add("hi");
myArrayList.add("there");

// Access elements by index (starting at 0!)
for (int i = 0; i < myArrayList.size(); i++) {
    println(myArrayList.get(i));
}

// hi
// there
```



Methods in the `ArrayList` Class

`boolean add(<T> element)`

Adds a new element to the end of the `ArrayList`; the return value is always `true`.

`void add(int index, <T> element)`

Inserts a new element into the `ArrayList` before the position specified by `index`.

`<T> remove(int index)`

Removes the element at the specified position and returns that value.

`boolean remove(<T> element)`

Removes the first instance of `element`, if it appears; returns `true` if a match is found.

`void clear()`

Removes all elements from the `ArrayList`.

`int size()`

Returns the number of elements in the `ArrayList`.

`<T> get(int index)`

Returns the object at the specified index.

`<T> set(int index, <T> value)`

Sets the element at the specified index to the new value and returns the old value.

`int indexOf(<T> value)`

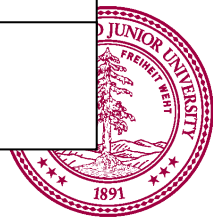
Returns the index of the first occurrence of the specified value, or `-1` if it does not appear.

`boolean contains(<T> value)`

Returns `true` if the `ArrayList` contains the specified value.

`boolean isEmpty()`

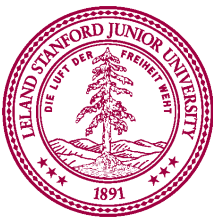
Returns `true` if the `ArrayList` contains no elements.



ArrayLists + Primitives =

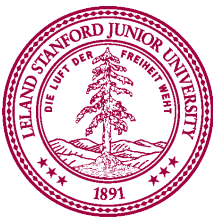
```
// Doesn't compile ☹️  
ArrayList<int> myArrayList = new ArrayList<int>();
```

ArrayLists can only store objects!



ArrayLists + Primitives =

Primitive	“Wrapper” Class
<code>int</code>	<code>Integer</code>
<code>double</code>	<code>Double</code>
<code>boolean</code>	<code>Boolean</code>
<code>char</code>	<code>Character</code>



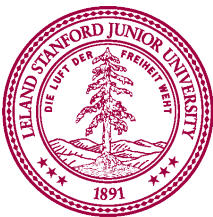
ArrayLists + Wrappers =

```
// Just use wrapper class when making an ArrayList
ArrayList<Integer> numList = new ArrayList<Integer>();

numList.add(123);
numList.add(546);

int firstNum = numList.get(0);    // 123
int secondNum = numList.get(1);  // 456
```

Conversion happens automatically!



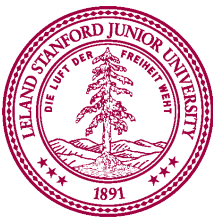
Demo: File Processing + ArrayLists

```
ArrayLists_OpeningCrawlSoln
```

```
    Karel Wars Episode VII:  
    Revenge of the SuperKarel
```

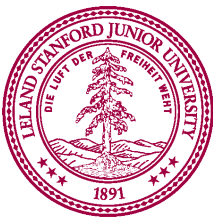
```
    An evil SuperKarel is threatening to  
    disturb the grid world once again.  
    What was once a peaceful planet has  
    now been thrown into chaos.
```

```
    By taking the turnRight() command  
    hostage, SuperKarel has forced all  
    other Karels to turnLeft() three times.
```



Summary of Today

- BufferedReaders let us read a file line by line.
- Since bad things can happen outside our control when reading files, we must use try/catch.
- ArrayLists are homogeneous lists of objects.
- You can add, remove, get, find, etc. on ArrayLists.



ArrayList

Good luck on the
midterm!

