A Picture is worth $2^{10}$ words
PFTD

● **Provide review and examples that review concepts for NBody from programming standpoint**
  - Scanner, File, java.util and java.io
  - Exceptions and handling or throwing
  - public static void main and static/non-static methods
  - Objects, instance variables, this

● **Some background on logistics**
  - Starter code and git, Ambient, copy/paste
  - The good, bad, and ugly of Eclipse

● **Further preparation for next week**
  - Tradeoffs and Optimizations
  - WOTO and working solo
What's the Point?

● Point a = new Point(); Point b = a; b.x = 41; a.x ?
  ➢ All variables are values, value of object is a reference
  ➢ Assignment to a variable copies the value

```java
class Point {
    double x;
    double y;
    public Point() {
        x = y = 0;
    }
    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }
}
```
javarepl.com  Check Understanding

- Revisit questions now that you've seen arrays, but do NOT use javarepl.com in answering them now


- Object variables are points to the object
- Strings are immutable
- int value converted to Integer in ArrayList (and vice versa) as needed
Java packages

● A package is a coherent grouping of related classes
  ➢ Share some characteristics in common
  ➢ Corresponds to a folder, but conceptual not physical
  ➢ What do java.net, java.io, java.security group together?

● The java.lang package is always accessible in code
  ➢ Integer, System, Exceptions, language constructs

● Other packages (or classes) explicitly imported
  ➢ Use java.util.* or java.util.Scanner, advantages?
How do you use/understand classes?

- Use the public API, (examples & documentation)
  - Application Programming Interface
  - Concept widely used across all languages and systems
  - Use the Twitter API, use the IMDB API, use the ...
  - What's ola's API story?

- Documenting your own API/classes, good idea?
  - Easy answer, other answers
  - Should we adopt industry practices?
- https://google.github.io/styleguide/javaguide.html
Eclipse from Scratch

● How do we read from a file? Specify file?
  - What is a File? What is a file
  - FileSelector.java see examples

● What is a Scanner? An InputStream?
  - Packages java.util and java.io, but Eclipse knows!!

● How do you solve a problem like an Exception?
  - Catch in some situations, won't have them really
  - Rethrow the exception, propagate with good messages!
YAWC or UWA: Unique Words Again

- We'll use ArrayList, .add, and .indexOf or .contains
  - Slow, but simple
  - Build on smart and more complicated

- We'll use HashSet, .add
  - Let the set keep track of avoiding duplicates
  - Hashing is very fast

- Both examples could be used in same method!
  - Both implement Collection and Iterable (interfaces)
Fran Allen

- IBM Fellow, Turing Award
  - Optimizing compilers
- Taught high school for two years, then Master's degree and IBM
  - Teachers excited me to learn

I've always felt that theory without practice is maybe nice and maybe pretty, but it’s not going to influence computing as much as the practice side. But the practice has to be backed up with the ability to talk about it, reason about it, and formulate it so that it can be reproduced.
YAWC example: summary

- Use Eclipse to help you write code
- Use the Java API to help you write code
- Use previous examples to help you write code
- Use online resources to help you write code
- Use course resources to help you write code

- Use your own judgment, experience, and abilities to help you write code
Expanding YAWC program

- Let's actually count the word occurrences
  - Illustrate several features of Java we'll use frequently
  - Expand on using this in code
  - Discuss getters (and setters), keeping data private

- Why is WordCounter like a POJO? Different?
  - Look at constructor, getter
  - Look at modifier method
  - Look at .toString() and then look at .equals()

https://git.cs.duke.edu/201fall16/classwork/tree/master
Questions about Classes

- Looking at WordCounter.java to understand some aspects of programming in Java
  https://git.cs.duke.edu/201fall16/classwork/blob/master/src/WordCounter.java


- Understand constructors, .toString(), .equals(..)
Sergey Brin

- Simple ideas sometimes can change the world [wikipedia]
  - Works because of scale
  
http://www.bloomberg.com/video/66114966/

- Co-created pagerank (Larry Page), which evaluates links to a page and the importance of those links, based on the importance of the page from which the links come which...

http://upload.wikimedia.org/wikipedia/commons/0/00/Sergey_Brin_2008.jpg
Practice Solving a Problem

● One method for choosing a "best" guess in Jotto
  ➢ Recall that computer removed all words that couldn't be the secret/hidden word
  ➢ If guess "stork" has zero letters in common, keep only those words that have zero letters in common with stork

● Which of those left should we choose as guess?
  ➢ Shuffle and pick any
  ➢ Choose the word that is most likely to remove the most words no matter what user responds. What???

● When is mathematics useful in computer science?
Expected Value

- Consider the word "climb", a candidate guess
  - User could respond 0, 1, 2, 3, 4, 5 (ignore 6 for now)
  - If user responds 0 – how many words left?
  - If user responds 1 – how many words left?

- Suppose we can calculate
  - How?
  - Software 😊

<table>
<thead>
<tr>
<th># common</th>
<th>Words remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>83</td>
</tr>
<tr>
<td>3</td>
<td>77</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>
What is the score of "climb"

- Total # words = $100 + 200 + 83 + 77 + 20 = 480$
  - Probability of user reply: 0 letters in common $100/480 = .208$
  - Expect to remove 100 words with probability 0.208

Expected value = $131.416$, expectation for "climb"

\[(100 \times .208) + (200 \times .416) + (83 \times .172) + (77 \times .160) + (20 \times .041)\]

<table>
<thead>
<tr>
<th># common</th>
<th>Words remaining</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>.208</td>
</tr>
<tr>
<td>1</td>
<td>200</td>
<td>.416</td>
</tr>
<tr>
<td>2</td>
<td>83</td>
<td>.172</td>
</tr>
<tr>
<td>3</td>
<td>77</td>
<td>.160</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>.041</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>