Compsci 201
Work, Nbody, ArrayLists

jshell> String[] a = {"ant", "bat", "cat", "dog"}
a == String[4] { "ant", "bat", "cat", "dog" }

jshell> System.out.println(a)
[Ljava.lang.String;@5b275dab

jshell> System.out.println(Arrays.toString(a))
[ant, bat, cat, dog]

Susan Rodger
January 29, 2020
F is for ... 

- Folder
  - aka Directory – where things are stored in Git

- Function
  - Abstraction – a method in Java
PFTW

• Getting things done in 201
  • How to succeed and enjoy the effort

• Mundane Java-isms
  • From char to autoboxing: primitives
  • What is this?

• Generic classes: How ArrayList works
  • Design, create, test, measure
Getting Things Done in 201

• What do these data mean for you, for me, for the community of 286 students in Compsci 201?

<table>
<thead>
<tr>
<th>Choose the APT</th>
<th>&lt; 1 hour</th>
<th>1-2 hours</th>
<th>2-4 hours</th>
<th>4-6 hours</th>
<th>6-10 hours</th>
<th>&gt; 10 hours</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessLevel</td>
<td>241</td>
<td>20</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>266</td>
</tr>
<tr>
<td>CirclesCountry</td>
<td>160</td>
<td>81</td>
<td>20</td>
<td>4</td>
<td>1</td>
<td></td>
<td>266</td>
</tr>
<tr>
<td>Common</td>
<td>94</td>
<td>38</td>
<td>14</td>
<td>4</td>
<td></td>
<td></td>
<td>150</td>
</tr>
<tr>
<td>DNAMaxNucleotide</td>
<td>112</td>
<td>97</td>
<td>36</td>
<td>14</td>
<td>3</td>
<td>3</td>
<td>265</td>
</tr>
<tr>
<td>Gravity</td>
<td>254</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td>261</td>
</tr>
<tr>
<td>SandwichBar</td>
<td>109</td>
<td>100</td>
<td>36</td>
<td>13</td>
<td>1</td>
<td></td>
<td>259</td>
</tr>
<tr>
<td>Totality</td>
<td>236</td>
<td>27</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>267</td>
</tr>
<tr>
<td><strong>Grand Total</strong></td>
<td><strong>1206</strong></td>
<td><strong>367</strong></td>
<td><strong>117</strong></td>
<td><strong>35</strong></td>
<td><strong>6</strong></td>
<td><strong>Total</strong></td>
<td><strong>267</strong></td>
</tr>
</tbody>
</table>

1/29/2020 Compsci 201, Spring 2020 7
From Last Time …
Go over
WOTO: Correctness Counts

Object class, equals method

• In JavaDoc

• Signature of equals method

```java
public boolean equals(Object obj)
```
@Override equals

• Create a new Point method
  • Use annotation @Override, help with errors
  
  ```java
  boolean equals(Object o) {
    …
  }
  ```

• Must use this signature, to implement:
  • Cast parameter appropriately
  • Compare instance fields
Point inherits Object.equals

• This doesn’t work for Point objects!
  • Default simply uses ==, no idea about points
  • a.equals(b) if a and b reference the same object
  • Two different (0,0) points not the same
Point equals fixed!

```java
@Override
public boolean equals(Object o) {
    Point other = (Point) o;
    if (other.myX == myX && other.myY == myY) {
        return true;
    }
    return false;
}
```
Contract for Equality

- Reflexive \texttt{x.equals(y)} then \texttt{y.equals(x)}
- Transitivity: \texttt{x.eq(y), y.eq(z)} then \texttt{x.eq(z)}

- Check \texttt{x.equals(x)} as a special case with ==
- Check \texttt{this.getClass() == o.getClass()}
  - Don’t want to have an apple == orange

- Cast Object parameter and use instance variables
  - See Point as example
Amanda Randles, Duke 2005

- ACM Grace Murray Hopper Award (<= 35 yo)

For developing HARVEY, a massively parallel circulatory simulation code capable of modeling the full human arterial system at subcellular resolution and fostering discoveries that will serve as a basis for improving the diagnosis, prevention, and treatment of human diseases.

//XXX and Amanda Peters
//Compsci 100: Huffman Coding
//November 19, 2002

I felt like working in a pair was a really successful way to complete the program. It helped the most when it came to working out basic logic and finding errors. I found it really helpful because he often would see the basic logic to the code and I could help more with the implementation. I feel like it was a successful group and we both contributed a lot.
Reading Points

• We'll typically use a Scanner to read values
  • Use `.hasNext()`, `.hasNextDouble()`, ...
  • If/while there's more to read? Call `.next()`

• Method `.next()` returns a String
  • Method `.nextDouble()` returns a double …

• See PointReader.java class, useful in NBody
Scanner Sources for Reading

• Construct a Scanner from System.in
  • Reads from keyboard/console
  • `.hasNextX()` true until end-of-file OR no X
    • Control-D on OS X, Control-Z on Windows

• Construct a Scanner from a File
  • Reads from file, *exception could happen*
  • `.hasNextX()` true until all of file read OR no X
    • Each call of `.nextX()` returns the next X, internally the Scanner "remembers" where it last read
Scanner hasNext and next

- Think about scanner as a long reel/source of data
  - If `hasNext()` returns true, there is something to read by Scanner cursor/reader
  - Calling `next()` returns *and advances* cursor
  - Scanner object maintains cursor internally
- Source: file, String, terminal, ..
N-Body Simulation

- **Class CelestialBody represents Celestial Body**
  - Planet, Sun, Asteroid
  - Models an object in 2D space, not 3D
  - Position, Velocity, Mass, Image for display

- **Class NBody drives the simulation**
  - Compute gravitational forces: physics
  - Time-step simulation
    - compute all forces, update, display
Class CelestialBody

• Illustrates standard Java idioms
  • Constructors, Methods, Instance Variables
• State is private: six instance variables
  • myXPos, … using my convention - this object
  • Initialized by constructors
• Methods are public
  • Include accessor aka getters for state
  • No setters, cannot change myXPos other than via the update method, a mutator
The Object Concept

• Every instance variable and every non-static method accessed/called after Object.Dot
  • `b.getX()`, `b.calcForcExertedBy(other)`

• From within a class, e.g., `CelestialBody`
  • `myXPos`, `getX()`, `this.myXPos`,
  • All are equivalent as is `this.getX()`

• Some prefer always using this. – clearer?
NBody numbers

• Floating point issues, problems, quandaries
  • When is \((a + b) + c \neq a + (b + c)\)
  • When is \(\frac{a}{b} \times c \neq \frac{a \times c}{b}\)
    • Watch for this in Gradescope tests!!
Debugging Arithmetic

• Order of operations with floating point values can result in overflow, underflow, more
  • Small number + Big number …

```
jshell> (5 + 1e20) + -1e20
$33 ==> 0.0

jshell> 5 + (1e20 + -1e20)
$34 ==> 5.0
```
Debugging double Arithmetic

• Integer values are not the same as Double values
  • 1/0 is … whereas 1.0/0 is …

```java
jshell> Double.MAX_VALUE
$48 == 1.7976931348623157E308

jshell> Double.MAX_VALUE + 2
$49 == 1.7976931348623157E308

jshell> Double.MAX_VALUE * Double.MAX_VALUE
$50 == Infinity

jshell> 1.0/0
$51 == Infinity

jshell> Math.sqrt(Double.POSITIVE_INFINITY)
$52 == Infinity

jshell> Double.POSITIVE_INFINITY + Double.NEGATIVE_INFINITY
$53 == NaN

jshell> 1/0
   | Exception java.lang.ArithmeticException: / by zero
   | at (#54:1)
```
Completing NBody

• Please read the TL;DR document
  • Test at each step, push constantly using Git

• After using supplied Test… classes, proceed to simulation
  • Must be able to read data file to simulate
  • Understand the basics, read carefully

• Analysis: complete before submitting to Gradescope for final submission
Now look at DNAMaxNucleotide

• Return the strand from strands array with most occurrences of nucleotide nuc. Return longest such strand

```java
public class DNAMaxNucleotide {
    public String max(String[] strands, String nuc) {
        // fill in code here
    }
}
```

• Example

2. strands = {"agt", "aagt", "taattt", "ccatc" }  
   nuc = "g"

Returns: "aagt" since both "aagt" and "agt" have one occurrence of 'g', but "aagt" is longer.
Algorithm - DNAMaxNucleotide

• Does this code make the algorithm clear?
  • Why must count be a helper method?
  • Why can't max = 0 before loop?

```java
public String max(String[] strands, String nuc) {
    String ret = "";
    int max = 1;
    for(String s : strands) {
        int nc = count(s, nuc);
        if (nc > max || (nc == max && s.length() > ret.length())) {
            ret = s;
            max = nc;
        }
    }
    return ret;
}
```
Two Versions of Helper Method

- Iterating over each character of a string
  - Note that `nuc` is a one-character string
  - How does `s.substring(a,b)` work?

```java
private int count(String s, String nuc) {
    int total = 0;
    for(int k=0; k < s.length(); k+= 1) {
        String one = s.substring(k,k+1);
        if (one.equals(nuc)) {
            total += 1;
        }
    }
    return total;
}
```
Critique of another implementation

• Where does this solution come from?
  • Strings are immutable, s.replace(…)
    • Replace every "a" with "" (nothing)

```java
private int count(String s, String nuc) {
    int tot = s.length() - s.replace(nuc, replacement: "").length();
    return tot;
}
```
WOTO

Donald Knuth

- aka “The Donald”
- Turing award (and others)
- Author of “The Art of Computer Programming”
  - Arguably most important book written in Computer Science
  - First publication: Mad Magazine

If you optimize everything you will always be unhappy.

Everyday life is like programming, I guess. If you love something you can put beauty into it.

https://www.youtube.com/watch?v=cK7yyjXfbc4
From Array to ArrayList

- Have int[], String[], CelestialBody[]
  - Array of any type, **but doesn't grow**
  - Can't use `.contains` with array, can't print
- The java.util.Arrays class has some help

```java
jshell> String[] a = {"ant", "bat", "cat", "dog"}
a ==> String[4] { "ant", "bat", "cat", "dog" }

jshell> System.out.println(a)
[Ljava.lang.String;@5b275dab

jshell> System.out.println(Arrays.toString(a))
[ant, bat, cat, dog]
```
java.util.ArrayList

- Growable array with many useful methods
  - [https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/List.html](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/List.html)
  - Can only contain Object types (no primitives)

- Convert from array?
  - **Arrays.asList**
    - It's a List!
  - String yes, int no

```java
jshell> ArrayList<String> b = new ArrayList<>();
b ==> []
jshell> b.add("ant")
$5 ==> true
jshell> b.add("bat")
$6 ==> true
jshell> b.add("cat")
$7 ==> true
jshell> b.size()
$8 ==> 3
jshell> System.out.println(b)
[ant, bat, cat]
jshell> b.indexOf("cat")
$10 ==> 2
jshell> b.indexOf("dog")
$11 ==> -1
```
From Array to ArrayList

• Can make conversion with Object, e.g., String
  • Use Arrays.asList as a bridge, be careful

```java
jshell> String[] a = {"cat", "dog"}
a ==> String[2] { "cat", "dog" }

jshell> ArrayList<String> b = new ArrayList<>((Arrays.asList(a))
b ==> [cat, dog]

jshell> b.add("fox")
$23 ==> true

jshell> b
b ==> [cat, dog, fox]
```
Primitive Array? do it yourself

- No bridge from `Arrays.asList` since primitive
  - Loop and use autoboxing/unboxing
  - Conversion of `int` to `Integer` and *vice versa*

```java
jshell> int[] a = {1,2,3,4,5}
a ==> int[5] { 1, 2, 3, 4, 5 }

jshell> ArrayList<Integer> b = new ArrayList<>();
b ==> []

jshell> for(int val : a) b.add(val)

jshell> b
b ==> [1, 2, 3, 4, 5]

jshell> b.add(55)
$29 ==> true

jshell> b
b ==> [1, 2, 3, 4, 5, 55]
```
Objects, Primitives, Arrays/Lists

• array can hold any type: `int[]`, `String[]`
• `ArrayList` only Object types, not primitives
  • Autoboxing allows for add/get `int` :: `Integer`
• `ArrayList<Object> a, a.toArray(…)` array
  • Syntax is not intuitive, see examples in code
• `Arrays.asList(Object[])` to `ArrayList`
  • Actually returns `List`, not `ArrayList`, …
LeetCode -- real world APT?

- [https://leetcode.com/problems/unique-morse-code-words/](https://leetcode.com/problems/unique-morse-code-words/)
  - "a" > ".-", "b" > "-..."
  - "z" > "--.."
  - Note "gin" > "--...-." and "zen" > "--...-.

- Given an array of strings, how many unique encodings are there?
  - Also given `String[]` of 26 Morse codes, where `code[0] = ".-"` for "a"
High Level Ideas

• First step: what algorithm/method will you use?
  • Verify that it's correct. High level isn't easy

• Is it necessary to look at/process every string?
• What value is returned, how to determine value?

• Stop, think, don't code, …
• What's a high level solution using known tools?
  • What is the method `makeMorse()`?
  • Talk to your interviewer … it's a dialog

```java
public int uniqueMorse(String[] words) {
    HashSet<String> set = new HashSet<>();
    for (String s : words) {
        set.add(makeMorse(s));
    }
    return set.size();
}
```
From Nothing to Done

• Basic ideas: how do we access encodings in an array where code[1] is for 'b', "-..."
  • Arithmetic with char values, 'b' – 'a' == 1
  • What about (int) 'b' == 97?
    • https://youtu.be/xLpfbcXTeo8?t=49

• Loop over characters in a String?
  • Index k with s.charAt(k)
  • Or for(char ch : s.toCharArray())

WOTO with Live/Leet Code

- Ideas for solving LeetCode problem
  - Given array of Strings, return number of unique Morse code encodings
  - How is a set useful here? Doable without?
From DNAMax to Morse Code

• loop on 18-23, why does ch-'a' serve as index?
  • Primitive char is an int except when printed

```java
private String makeMorse(String s) {
    String[] m = {
        "-", "--", "---", "--.", ".--", ".", "-.", "--", "-.", "-.", "-.", "-.", "-.", "-.", ".-", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", "-.", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".”,
    
    String ret = "";
    for(char ch : s.toCharArray()) {
        int dex = ch-'a';
        ret += m[dex];
    }
    return ret;
}
```
ArrayList<...>

- Generic aka parameterized type
  - Any Object subtype can be in ArrayList<..>

- Integer, Double, Char, Boolean are wrapper classes for primitives
  - Mostly these work. But immutable. Cannot increment an Integer, can create new one