CS 211: Introduction

Chris Kauffman

Week 1-1
Goals

- Motivate: Java and OO
- Problem Solving
- Course Mechanics
- Tools
Why Java / OO

Object-Oriented Design

- Method to decompose problems for solution (by computer programs)
- Style of programming / arranging code
- A way of thinking independent of any particular PL

Java

- Built around OO methodology
- Garbage collection
- Huge and useful standard library
- Tons of 3rd party libraries
- Widely deployable
- Huge corporate backers
- Fashionable methodology
- Will help get you a job
Java’s Place in the World

Pure Abstraction

- Python, JS
- Ruby, Shell
- C++, D
- Assembly
- VHDL
- Bread Board
- Gates
- Prolog, Lisp
- ML, Haskell
- Binary Opcodes
- Wires
- Electrons
- Java
- C
- Basic language mechanisms are still C-spirited
- Allows single dynamic dispatch

- Closer to hardware than some contemporaries
- "Hardware" is a Virtual Machine that is reasonably close to actual machines
- Abstracts away memory management (woot!)

Img Source: http://bpmredux.wordpress.com/
Kauffman’s Philosophy

- You will not program in java your entire career
- Object-oriented programming... 
  - Not well defined
  - Not widely understood
  - Has not made programs much better
  - May go out of style
- Problem solving will always be fashionable
- Good Program Design transcends time

Programming is

- Formalizing a solution method
- Translating into a mechanically executable form

Designing is

- Ensuring the solution can be implemented correctly
- Ensuring the solution can adapt to changes
Programmers Are Like Super Heroes

OO will be one tool in your growing utility belt. Make sure it’s not the only one.
Let’s Solve a Problem (or at least start)

- Each GTA is assigned some lab sections for CS211
- Each lab section has a day/time and a student count

Your program (pseudocode / python/ whatever)

1. Read the data file provided
2. Compute the Head Count for each GTA
3. Print the Discrepancy: difference between biggest and smallest head count

Input Data File

<table>
<thead>
<tr>
<th>#</th>
<th>Day</th>
<th>Time</th>
<th>Room</th>
<th>Count</th>
<th>GTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>R</td>
<td>12:30</td>
<td>4457</td>
<td>19</td>
<td>Raj</td>
</tr>
<tr>
<td>202</td>
<td>R</td>
<td>01:30</td>
<td>4457</td>
<td>15</td>
<td>Raj</td>
</tr>
<tr>
<td>203</td>
<td>R</td>
<td>02:30</td>
<td>5358</td>
<td>6</td>
<td>Adam</td>
</tr>
<tr>
<td>204</td>
<td>R</td>
<td>03:30</td>
<td>4457</td>
<td>21</td>
<td>Raj</td>
</tr>
<tr>
<td>205</td>
<td>R</td>
<td>11:30</td>
<td>5358</td>
<td>11</td>
<td>Ruoxi</td>
</tr>
</tbody>
</table>

Head Counts

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Raj</td>
<td>55</td>
</tr>
<tr>
<td>Adam</td>
<td>6</td>
</tr>
<tr>
<td>Li</td>
<td>11</td>
</tr>
</tbody>
</table>

Discrepancy: 49
### Full Data

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Head Counts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Discrepancy: 24</td>
</tr>
</tbody>
</table>

#### Input

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>R</td>
<td>12:30</td>
<td>4457</td>
<td>19</td>
<td>Raj</td>
</tr>
<tr>
<td>202</td>
<td>R</td>
<td>01:30</td>
<td>4457</td>
<td>15</td>
<td>Raj</td>
</tr>
<tr>
<td>203</td>
<td>R</td>
<td>02:30</td>
<td>5358</td>
<td>6</td>
<td>Adam</td>
</tr>
<tr>
<td>204</td>
<td>R</td>
<td>03:30</td>
<td>4457</td>
<td>21</td>
<td>Raj</td>
</tr>
<tr>
<td>205</td>
<td>R</td>
<td>11:30</td>
<td>5358</td>
<td>11</td>
<td>Li</td>
</tr>
<tr>
<td>206</td>
<td>R</td>
<td>12:30</td>
<td>5358</td>
<td>22</td>
<td>Li</td>
</tr>
<tr>
<td>207</td>
<td>R</td>
<td>01:30</td>
<td>5358</td>
<td>15</td>
<td>Li</td>
</tr>
<tr>
<td>208</td>
<td>R</td>
<td>02:30</td>
<td>5358</td>
<td>12</td>
<td>Adam</td>
</tr>
<tr>
<td>2H1</td>
<td>R</td>
<td>03:30</td>
<td>5358</td>
<td>17</td>
<td>Adam</td>
</tr>
<tr>
<td>210</td>
<td>R</td>
<td>01:30</td>
<td>1505</td>
<td>25</td>
<td>Adam</td>
</tr>
<tr>
<td>212</td>
<td>R</td>
<td>02:30</td>
<td>1505</td>
<td>24</td>
<td>Le</td>
</tr>
<tr>
<td>213</td>
<td>R</td>
<td>03:30</td>
<td>1505</td>
<td>24</td>
<td>Le</td>
</tr>
<tr>
<td>214</td>
<td>F</td>
<td>08:30</td>
<td>4457</td>
<td>3</td>
<td>Kacem</td>
</tr>
<tr>
<td>215</td>
<td>F</td>
<td>09:30</td>
<td>4457</td>
<td>10</td>
<td>Kacem</td>
</tr>
<tr>
<td>216</td>
<td>F</td>
<td>10:30</td>
<td>4457</td>
<td>12</td>
<td>Kacem</td>
</tr>
<tr>
<td>217</td>
<td>F</td>
<td>11:30</td>
<td>4457</td>
<td>24</td>
<td>Kacem</td>
</tr>
</tbody>
</table>
Variants

Feasibility
Determine if a GTA assignment is possible/impossible (Is the following assignment possible?)

- 201 R 12:30 4457 19 Li
- 202 R 01:30 4457 15 Raj
- 203 R 02:30 5358 6 Adam
- 204 R 03:30 4457 21 Raj
- 206 R 12:30 5358 22 Li

Balance
Determine the GTA assignment that is possible and has the lowest discrepancy
Tools

The official java tools of the course are

- **jdk 1.8**, official build and run tools from Oracle
- **DrJava**, a simple, superior java IDE (if you’re into IDEs)

Minor support given for (though not official)

- **jGrasp**, a decent IDE with drawing capabilities, used for some in-class examples

Special Note:

- I do not know how to use eclipse
- I will not be learning how this semester.
- If I can help it I will never learn eclipse.
- TAs may be able to help you but are not required to do so.
- In class I will use jGrasp, Emacs, and command line.
- If you have questions on those I’m happy to help.
Tools

DrJava

Eclipse/Netbeans
Alternatives

Vim

Emacs
Tools that Grow

DrJava

Eclipse

Emacs
Special Note on DrJava

We’ve made some improvements at GMU

▶ Better test result printing
▶ Fixed debugger activation bug
▶ Unofficial, trying to get into main distrib
▶ **Strongly** encourage DrJava users to grab this version
▶ Download here: https://cs.gmu.edu/~kauffman/drjava/
Upcoming

Lab this week

- Set up jdk 1.8
- Install DrJava if desired
- Compile a few basic programs
- Submit lab code by 5:00pm Sunday

Project 1 will be finalized in the next day or so
This Week/Weekend

- Make sure you set up your environment and can compile programs
- **READ** Reges/Stepp Ch 1-4
  - Overview of procedural programming in Java
  - Good review, entertaining writer
- Practice It Exercises associated with the chapters