Administrative Details

- Class roster
- About me
- Handout: Class syllabus
- Class meeting time: MWF 10-10:50
- Lecture location: Griffin 6
- Lab: Wed 12-2 and 2-4
- Lab location: TCL 217a and 216
- Lab code?

Today's Outline

- Course Overview
- Java refresher

Why Take CS136?

- To learn about:
  - Data Structures
    - Common ways to store and manipulate data
  - Advanced Programming
    - Use structures to write programs that solve (interesting) problems
- Note to students who have not taken 134: If (after a few days) you feel you are not ready for 136, you can probably switch to 134. Come see me about this.

Goals

- Identify basic data structures
  - Examples?
  - list, stack, array, tree
- Implement these structures in Java
- Learn how to evaluate and visualize data structures
  - Linked lists and vectors both represent lists of items
  - Different representations of data
  - Different algorithms for manipulating/accessing/storing data
- Learn how to design large programs that are easy to modify, extend, and debug
  - Have fun!

Example Programs

- Find a way to drive from Williamstown, MA to San Diego, CA
- Find the shortest way to drive from Williamstown, MA to San Diego, CA
- Schedule a flight with the fewest layovers from Albany, NY to Beijing, China
- Schedule exams so no students have conflicts
Common Themes

1. Identify data for problem
2. Identify questions to answer about data
3. Design data structures and algorithms to solve questions correctly and efficiently (Note: not all correct solutions are efficient, and vice versa!)
4. Implement solutions that are robust, adaptable, and reusable

Example: Boggle

Course Outline

- Java review
- Foundations of programming
  - Vocabulary
  - Analysis tools
  - Recursion
  - Methodology
- Basic structures
  - Lists, vectors, queues, stacks
- Advanced structures
  - Graphs, heaps, trees, dictionaries

Why CS136 == Awesome

- CS134 teaches you the basic mechanics of programming
- CS136 teaches you how to unlock the power of computers to solve problems in CS, econ, math, physics, etc.
- Data representation
- Algorithms
- Abstraction

Syllabus

- How to contact me
  - Office: TCL 304
  - Office hours: MTh 1:30p – 3:00p
    (or anytime when I’m in my office…email me)
  - Phone: x4251
  - Webpage: http://www.cs.williams.edu/~jeannie
  - Email: jeannie@cs.williams.edu
  - Anonymous feedback: See form on webpage
- Textbook
  - Java Structures: Data Structures in Java for the Principled Programmer, v7 Edition (by Duane Bailey)
  - Pick it up from Amanda Turner in TCL 303

Honor Code and Ethics

- The student handbook describes the Honor Code and Computer Ethics guidelines.
- You should also know the CS Dept computer usage policy.
  - If you are not familiar with these items, please review them.
- We take these things very seriously…

Your Responsibilities

- Come to lab and lecture on time.
- Read assigned chapters before class and lab.
- Bring textbook to lab (or be prepared to use PDF)
- Come to lab prepared!!!
  - Bring design docs for program
  - Prof + several TAs = lots of attention for you—take advantage of this!
- Do NOT remain confused. Get help.
- Don’t cheat.
- Participate in class discussions.
Why Java?

• There are lots of programming languages…
  • C, Pascal, C++, Java, C#, Python
• Java was designed in 1990s to ease Internet programming
• Java is good because:
  • It’s easy (well, easier than predecessors like C++) to write correct programs
  • Object-oriented – good for large systems
  • Easy support for abstraction

This semester, we will…

• …NOT use BlueJ. :-)
• …focus more on structures and algorithms than on graphics, networks, etc. (we can always add graphics later)
• …use Emacs and Java.
• …compile from a terminal rather than by clicking a button.

Java Review

Java Review Goals

• Review the big ideas
• Use lab/book/TAs/web to fill in any gaps
• Don’t hesitate to ask me or the TAs for a refresher on any topic
• We’re going to go fast…interrupt anytime

Java

• Variable types
  • int, double, boolean, String, …
• Statements
  • int x = 3;
  • x = x + 2;
  • if (x > 3) { … } else { … }
  • while (x < 2) { … }
  • for (int i = 0; i < x; i++) { … }
• Comments
  • //this is a comment
  • /* so is this */

Sample Programs

• Hello.java
  • Write a program that prints “Hello” to the terminal.