Algorithms and OOD (CSC 207 2013F) : EBoards

CSC207.01 2013F, Class 06: Debugging

Overview

- Admin.
- Homework questions.
- Some notes on Java.
- A bit about debugging.
- Lab.
- Reflection (maybe).

Admin

- New partners!
- A friendly reminder that "Learning from Alumni" is a great opportunity to get real-world recommendations on how to prepare for a job or internship (and to network). And you get 1 credit. Thursdays, 2:15-4:05.
- I’m still working on getting HW1 graded. Time is just a limited commodity (as I think you all know from your own experience).
- Note: Sometimes the html eboards don’t get updated right after class. You can always see what I typed by going to the original source.
  - And send me a note if you see something missing.
- Reminder: Mentor session Wednesday night (8pm, I believe)
- Study break, tonight, 8pm, CS Commons
- When do you want me to distribute homework? Wednesday, *Tuesday*, Monday?
- Readings for Tuesday: Making Your Own Classes and Standard Object Methods
- EC opportunities:
  - Humanities Center Speaker Sarah Hendron, Wednesday, 7:30 p.m., JRC101 Waking the Machines: Art, Design, and Adaptive Technology
  - Learning from Alumni 2:15-4:05 Thursday: Ian Lunderskov ’08
  - Tentative Thursday Extra: Matt Atherton ’95
  - CS Table, Friday: Trusting Trust.
  - More?

Homework Questions

- Do we have to document our tests?
  - "Tests are self documenting"
  - But a few notes never hurt. See Friday’s eboard.
- Do we need inline comments
  - You should have written them first
  - If your code is understandable, less necessary

Some notes on Java

- Method names are almost always preceded by something and a period
  - Current stuff: ClassName.methodName(...)
  - Dealing with objects: objectName.methodName(...)
- Concrete examples
  - whatever = SampleMethods.removeAs("alphabet");
  - pen.println(whatever);
- Why don’t you use System.out? [Warning: Religious wars!]
  - System.out is easy.
  - System.out does not make for code that is easy to change If you change where you want stuff to go, you have lots of lines to change If we use an object, we can change the object in one place and it gets fixed everywhere If we use objects, we can distinguish kinds of output
    - Error/warning/"debugging" messages
    - Normal output
- I’m optimistic about your abilities - I assume you can figure out the small differences between C and imperative Java. When I’m wrong, talk to me or Lea.

A bit about debugging

- Almost everyone writes buggy code. We need to find ways to a. Identify that bugs exist b. Find where in the code the bugs occur
- One solution: Use printf
  - Debugging appears in the middle of your output
  - Adds programmer effort
  - Painful to add and remove printfs
  - Tons of output

Lab

Reflection (maybe)