

CSC207.01 2013F, Class 54: Patterns of Object and Algorithm Design

Overview

- Preliminaries.
 - Admin.
 - Questions on Homework 11.
 - Notes on the Final
- Algorithm design.
- Object design.
- Code design.
- ADT design.
- Data structure design.

Preliminaries

Admin

- Our model for most sections of today's class will be "talk for a few minutes as a small group, then share your answers with the class".
- Upcoming extra credit opportunities:
 - CS Extra: Multiple Models of Mediascripting
 - CS Table Friday: Timing code.

Questions on HW 11

How should we do unit tests or experiments?

Build a few simple objects and see if they have the correct fields.

```
JSONObject obj = JSONParser.parse("{\"name\":\"wheelie\"}");  
assertEquals("wheelie", obj.get("name"));
```

Should we deal with spaces, tabs, newlines, etc?

It would be nice. But it should be easy. Before you check a character, use

```
while ((i < str.length() && (Character.isWhitespace(ch = str.charAt(i))))  
    i++;
```

Or you could be smarter than me and delete all of the whitespace in the string.

The Final

- Available now in draft form.
- Some code available now. Remaining code available tomorrow.
- Time for questions rest of the week.

Algorithm design

Given an interesting problem, how do you get started?

Examples

- Reverse a linked list in place
- Find the median value in an unsorted array

Techniques

- Google / StackOverflow [assume that they fail you]
- Think about the problem in more depth:
 - What are the starting data? (preconditions)
 - What should the ending data look like? (postconditions)
 - Write lots of examples (tests) of input and output
- Think about similar problems we've done before
- Draw pictures
- If writing loops, think about invariants (often pictorial)
- Look up the portion of the class in which you studied the problem
- Work through it by hand and then generalize
- Write crappy code and hack at it until it works or until you understand

What Sam often does

- Try to solve it "by hand" to gain some insight
- Look for similar problems I've solved before
- Draw pictures
- Think about approaches that have worked in the past
 - Divide and conquer
 - Dynamic programming (if I solve a bunch of smaller problems, does that help?)
 - Greed - Given a series of choices, make the one that seems most valuable

Object design

Code design

Copyright (c) 2013 Samuel A. Rebelsky.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.