

CS1110 Stepwise refinement, wrapper classes 17 Feb

Thursday's lecture: no reading, *but be there or be square (or lost)!*

- Recursion can be a difficult topic, but we'll make it easy.
- We anticipate distributing Assignment 4 then.

Prelim: 7:30-9pm Thursday 19 Feb, Baker Lab 200

All students with conflicts have already been contacted about alternate arrangements.

• Review the "what you need to know for prelim 1" document and go through old prelims and solutions: <http://www.cs.cornell.edu/courses/cs1110/2009/exams/exams.html>

- Upcoming office hours: see <http://www.cs.cornell.edu/courses/cs1110/2009/exams/exams.html>
- Prof Gries: Thursday 12:45-1:45
- Prof. Lee is traveling Tuesday afternoon through the weekend. Sorry, but this week *no appointments with her and her Thursday office hour is canceled.*
- T.A.'s: some hours Tuesday and Wednesday afternoon
- Consultant hours 4-10pm today, Wednesday, Thursday in the ACCEL lab

1

Best. Study tip. Ever.

Cornell's Learning Strategies Center posts a lot of great information on study skills, taking exams, time & stress management, etc.

<http://lsc.sas.cornell.edu/SSWorkshops/SKResources.html>

Every day after classes, retrieve your notes ...

--- you *do* take notes, don't you? ---

...and read them over.

This takes very little time, and yet:

- a) really makes material "stick" in one's mind, and
- b) helps you figure out what you don't understand early on, so you can get it straightened out faster.

This was a real game-changer for me.

2

More stepwise refinement, or, lies, damned lies, and statistics

James Pennebaker et al., "Lying words: predicting deception from linguistic styles", 2003:

Claim: deceptive communication is characterized by:

- fewer 1st-person singular and third-person pronouns ("I", "they")
- more negative emotion words ("hate", "enemy")
- fewer "complexity" words ("but", "except", "without")

I am doing research using computational-linguistics techniques to determine: are these the best cues? Can we build better models for linguistic lie detection?

Joint work with CS grad student Cristian Danescu Niculescu-Mizil and CS undergrad Haden Lee, in consultation with Comm. Prof. Jeff Hancock.

3

Motivations (besides automatic lie detection being inherently cool)

• Interdisciplinary project: computer science, psychology, linguistics, and even politics, as we shall see:

• Data: the Iraq War Card False Statements Database

<http://projects.publicintegrity.org/WarCard/Search/Default.aspx>

• Additional demonstration that you can already do interesting things (even research!) with what you know now.

• stepwise refinement, *writing and reading specs carefully*, String manipulation, and recursion

Caveats: Multiple solutions are possible. We'll also earn material later in the course that leads to even better solutions. (That's part of why we'll talk about wrapper classes at the end of today.)

4

A task formulation

Given

- a source file with known-false-statement ("lie") mark-up
- a target word *w*

Report these two statistics:

- number of hits --- occurrences of *w* in a "lie"
- number of misses --- occurrences of *w* in a "non-lie"

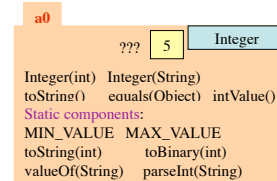
1. Find good marked-up data source
2. Get data from source into String format
3. Process the occurrences of *w*
 - We can use our StockQuote know-how on a single occurrence.
 - **How on earth can we account for multiple occurrences of *w*?**

5

Wrapper classes. Read Section 5.1 of class text

Soon, we'll wish to deal with an **int** value as an object.

"Wrapper class" Integer provides this capability.



An instance of class Integer contains, or "wraps", one **int** value.

You can't change the value. The object is *immutable*.

Instance methods: constructors, toString(), equals, intValue.

Static components provide important extra help.

6

Each primitive type has a corresponding wrapper class. When you want to treat a primitive value of that type as an object, then just wrap the primitive value in an object of the wrapper class!

Primitive type	Wrapper class
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

Each wrapper class has:

- Instance methods, e.g. equals, constructors, toString,
- Useful static constants and methods.

```
Integer k= new Integer(63);      int j= k.intValue();
```

You don't have to memorize the methods of the wrapper classes. But be aware of them and look them up when necessary. Use Gries/Gries, Section 5.1, and ProgramLive, 5-1 and 5-2, as references.

7