CS1110. Lecture 2, 28 Jan 2010. Objects & classes

Reading for this lecture: Section 1.3. It's most important that you **study this section over the weekend** and **practice** what is taught using DrJava.

PLive: Activities 3-3.1, 3-3.2, 3-3.4 (not 3-3.3), 3-4.1, 3-4.2.

Summary of lectures: On course page, click on "Lecture summaries".

Reading for Tuesday, 8 Sep.
Sections 1.4, (p. 41); 13.3.1 (p. 376).

Quote for the day:
Computers in the future may weigh no more than 1.5 tons.
--Popular Mechanics, forecasting the relentless march of science, 1949

1

CMS: Developed by the CS Department. Java based.

If you have not been receiving emails from us, sent out from the CMS, then either:

1. You are not registered in the CMS. Email Maria Witlox mwitlox@cs.cornell.edu and ask her to register you. Needs your netid.

2. Your email is bouncing. Your Cornell system is not set up correctly or the place to which you forward us is having trouble. Best thing to do:

email yourself, at netid@cornell.edu, see what happens, and fix it.

AEWs
Sign up for the 1-credit AEW sections for CS1110.

Two hrs per week. Nothing else.

Not remedial.

Mon 7:30-9:25pm: 12 spots

Fri 2:30-4:25: 2 spots

2

Two aspects of a programming language

- Organization – structure
- Procedural — commands to do something

Example: Recipe book

- Organization: Several options; here is one:
  - Appetizers
    - list of recipes
  - Beverages
    - list of recipes
  - Soups
    - list of recipes
  - …
- Procedural: Recipe: sequence of instructions to carry out

Parts to this course

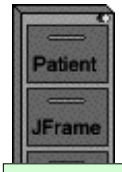
structural
objects
classes
methods
inheritance

procedural
assignment,
return,
if-statement
iteration (loops)
recursion

miscellaneous
GUIs
exception handling
Testing/debugging

3

A class is a file-drawer. Contents: manila folders, each containing the same kind of information



c1

class name: Patient

name: L. Lee
address: New York
owes: $250.00

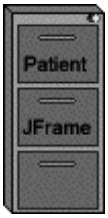
Name on tab (c1): anything you want, as long as it is unique

manila folder: an **object** or **instance** of the class

name, address, owes: **variables**, called **fields** of the folder

4

A class is a file-drawer. Contents: manila folders, each containing the same kind of information



c1

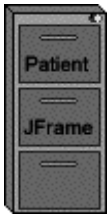
Patient

name: L. Lee
address: New York
owes: $250.00

Instructions to be carried out by different people:
change the name, get the name, bill the patient, receive money from patient, insert teeth xrays into the folder, …

5

A class is a file-drawer. Contents: manila folders, each containing the same kind of information



c1

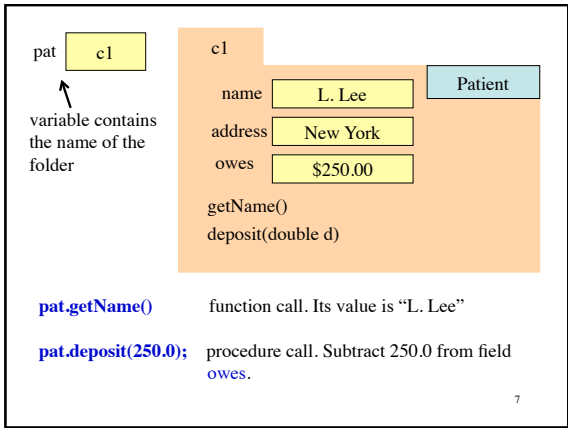
Patient

name: L. Lee
address: New York
owes: $250.00

getName()
deposit(double d)

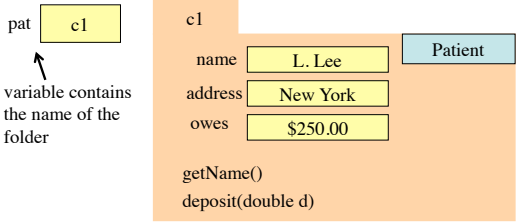
Instructions to be carried out by different people: methods.
getName is a **function**; it returns a value.
deposit is a **procedure**; it does some task, doesn't return value

6



pat.getName() function call. Its value is "L. Lee"

pat.deposit(250.0); procedure call. Subtract 250.0 from field owes.



new Patient() An expression: create a new folder (put it in file-drawer Patient) and give as the value of the expression the name of the folder.

pat= **new** Patient(); A statement: evaluate **new**Patient() and store its value (the name of the new folder) in variable pat.

variable contains name of folder

We will demo the use of most of these methods

An object (manila folder) of class

Javax.swing.JFrame

is associated with a window on your computer monitor. It has (among others) these functions:

getHeight() get Width() getX() getY()
getTitle() isResizeable()

and these procedures:

show() hide()
setTitle() setSize(int, int)
setLocation(int, int) setResizable(boolean)

In groups of 2, draw an object (manila folder) of this class, and put the name **a0** on its tab.

Comments from last semester

I understand classes and objects fairly well, and I thought the file drawer/file folder analogy was very helpful.

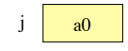
I think I'm definitely prepared for 2110. The folder/file drawer analogy was actually very helpful for a first-time Java programmer in understanding them.

I did learn the concept before coming to this class, CS1110 is really what made me understand how objects and classes work.

The folder was a great way to learn objects and classes. It simplified a very complex concept.

Teaching methods were terrible. … boxes and folders made the subject more confusing than it should be.

I'm still a bit dubious about the whole file folders and cabinets thing.



variable contains the name of the folder

- Read section 1.3.
- Practice what we did in class in DrJava.
- Try the self-review exercises on page 40.

```
j= new javax.swing.JFrame();
j.show();
…
```

Expression **new** JFrame()

Create new folder and put in file drawer JFrame.

Statement jf= **new** JFrame();

Create new folder, as above, and place its name in variable jf.

Thereafter, use

jf. *method-name* (*arguments, if any*)

to call methods of folder (object) jf.

package: A collection of classes that are placed in the same directory on your hard drive. Think of it as a room that contains file cabinets with one drawer for each class.

package **java.io** classes having to do with input/output

package **java.net** classes having to do with the internet

package **java.awt** classes having to do with making GUIs

package **javax.swing** newer classes having to do with GUIs

To reference class JFrame in package javax.swing, use:

javax.swing.JFrame

Instead: import javax.swing.*;

Then use simply JFrame