CS1110. Lecture 2, 4 Sep 2008. 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 "Handouts" and then "Outline of lectures held so far".

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

CMS: Course Management System.
Maintain grades, handle submitted assignments, post grades, handle regrades,etc. Developed by the CS Department. Java based.

From course web page, click on

cms.csuglab.cornell.edu/

Click on "sign in", enter cornell netid, password. Then, you will either be in the CMS and see the course description or you will see something like this:

CMS Overview
My Courses
CS1110 (student)

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

Two hrs per week. Nothing else.

Not remedial.

If you see CS1110, click on it. If not, email Maria Witlox, ask her to register you in the CMS for CS1110: mwitlox@cs.cornell.edu

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

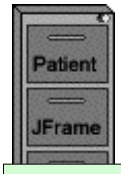
Parts to this course

structural
objects
classes
methods
inheritance

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

miscellaneous
GUIs
exception handling
Testing/debugging

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



c1

class name: Patient

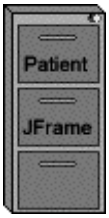
name	B. Clinton
address	New York
owes	$250.00

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

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



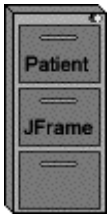
c1

Patient

name	B. Clinton
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, …

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



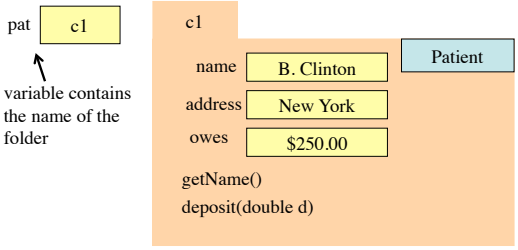
c1

Patient

name	B. Clinton
address	New York
owes	$250.00

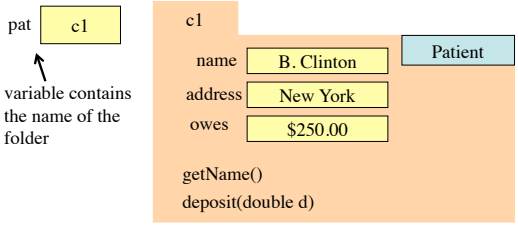
getName()
deposit(double d)

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



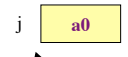
pat.getName() function call. Its value is "B. Clinton"

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.



An object (manila folder) of class

Javax.swing.JFrame

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

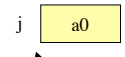
getHeight() get Width() getX() getY()
getTitle() isResizeable()

and these procedures:

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

We will demo the use of each of these methods

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



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 (jf should have first been declared).

Thereafter, use

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

to call methods of folder (object) jf.

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

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

The expression

new JFrame()

creates a new folder that goes in file drawer JFrame.

The statement

jf= **new** JFrame();

creates a new folder and places its name in variable jf (jf should have first been declared).

Thereafter, use

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

to call methods of folder (object) jf.

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