CS1130. Lecture 2, 27 Jan 2012. Objects & classes

Reading for this lecture: Section 1.3. **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".

See lecture on VideoNote

Reading for Tuesday, 7 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 Mech, forecasting the relentless march of science, 1949

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. Not registered in 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.

Quiz on Tuesday. Everyone should get 100.

1. What is a type?

2. How do you execute (carry out, perform) the assignment statement?

Mon 7:30-9:25pm:
Fri 2:30-4:25:

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

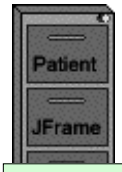
Parts to this course

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	L. Lee
address	New York
owes	$250.00

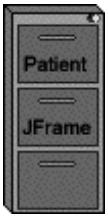
Name on tab (c1): anything you want, as long as it is unique

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

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

---

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



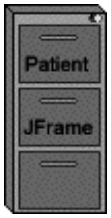
c1

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, …

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



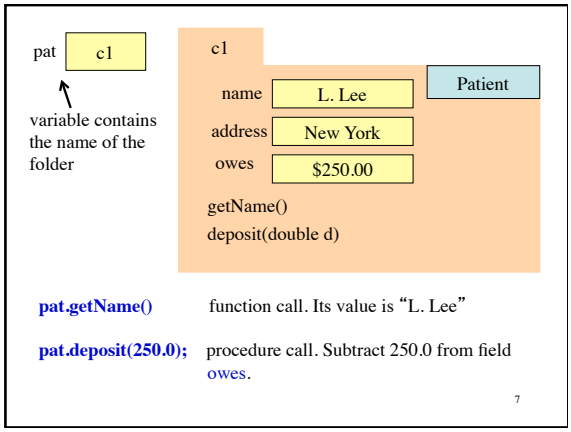
c1

Patient

name	L. Lee
address	New York
owes	$250.00

getName()
deposit(double d)

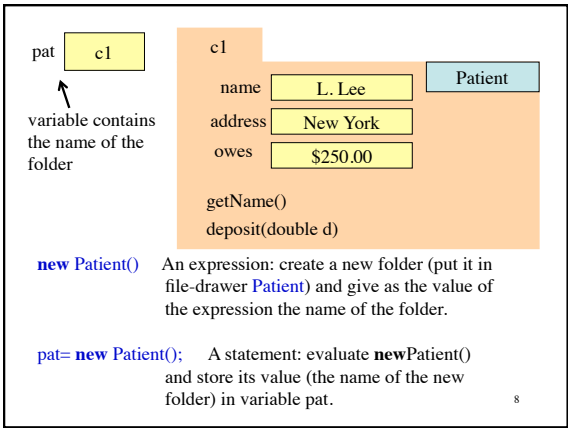
Instructions to be carried out by different people: methods.
getName is a **function**; it returns a value.
deposit is a **procedure**; it does some task, doesn't return value



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

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

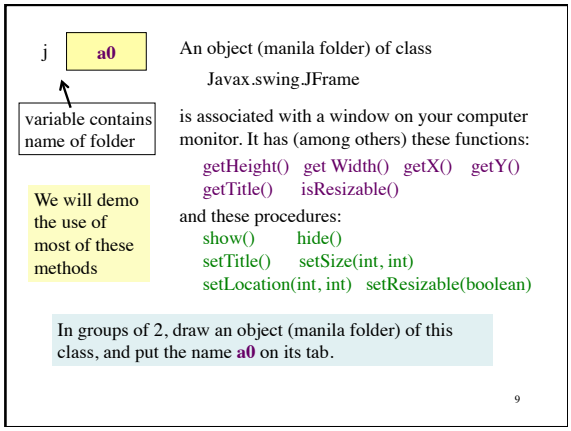
7



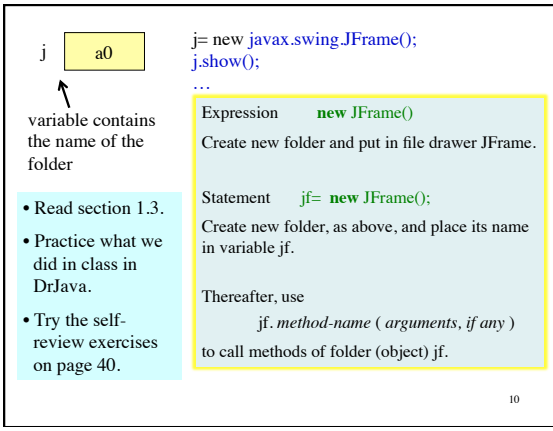
new Patient() An expression: create a new folder (put it in file-drawer Patient) and give as the value of the expression the name of the folder.

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

8



9



10

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

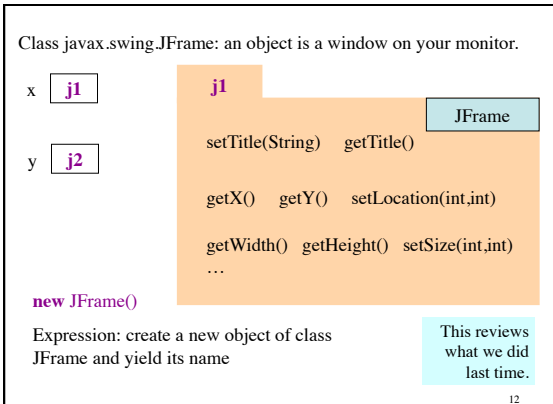
To reference class JFrame in package javax.swing, use:

javax.swing.JFrame

Instead: import javax.swing.*;

Then use simply JFrame

11



12

Class definition: The java construct that describes the format of a folder (instance, object) of the class.

```
/** description of what the class is for
  */
public class <class-name> {
    declarations of methods (in any order)
}
```

This is a comment

A class definition goes in its own file named

<*class-name*> . java

On your hard drive, have a separate directory for each Java program that you write; put all the class definitions for the program in that directory.

Class definition: The java construct that describes the format of a folder (instance, object) of the class.

```
/** description of what the class is for
  */
public class  C   extends <superclass-name>    {
    declarations of methods (in any order)

}
```

Class C has all the fields and methods that <*superclass-name*> does, in addition to those declared in C. Class C **inherits** the fields and methods of <superclass-name>.


```
/** description of what the class is for */
public class subclass-name extends superclass-name {
        declarations of methods
}
```

a0

superclass-name

methods and fields inherited from *superclass-name*

subclass-name

methods and fields declared in *subclass-name*

folder (object) belongs in file drawer for class *subclass-name*

First example of a procedure and of a function

```
/** description of what the class is for */
public class SQJFrame extends JFrame{
        /** Set the height of the window to the width */
        public void setHeightToWidth() {
                setSize(getWidth(), getWidth());
        }

        /** = the area of the window */
        public int area() {
                return getWidth() * getHeight();
        }
}
```



```
import javax.swing.*;
/** An instance is a JFrame with methods to square it and
    to provide the area of the JFrame */
public class SquareJFrame extends JFrame {
        declarations of methods
}
```

folder (object) belongs in file drawer for class SquareJFrame

To the left, draw a manila folder of class SquareJFrame. When we define methods, put them in the proper place

Javadoc

```
import javax.swing.*;
/** An instance is a JFrame with methods to square it and
    to provide the area of the JFrame */
public class SquareJFrame extends JFrame {
        /** = the area of the window */
        public int area() { … }

        /** Set the height equal to the width */
        public void setHeightToWidth() {…}
}
```

The class and every method in it has a comment of the form

/ specification */**

It is a Javadoc comment. Click on javadoc icon in DrJava to extract class specification. DO THIS AT LEAST ONCE IN LAB.

Javadoc

```
import javax.swing.*;
/** An instance is a JFrame with methods to square it and
    to provide the area of the JFrame */
public class SquareJFrame extends JFrame {
        /** = the area of the window */
        public int area() { … }

        /** Set the height equal to the width */
        public void setHeightToWidth() {…}
}
```

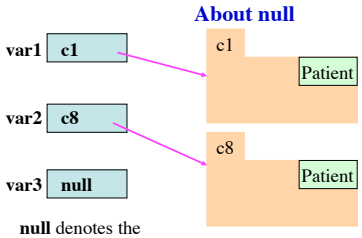
An object of class java.util.Date contains the date and time at which it was created.

It has a function toString(), which yields the date as a String.

Write a procedure setTitleToDate, which will set the title of the window to the date.

19

About null



null denotes the absence of a name.

var3.getName() is a mistake! You get a NullPointerException

20