CS100J 06 September 2005
The class definition

- **Course Management System for CS100J** is now populated with students who were pre-registered. Look at course web page to see how to get to it and what to do if you are not in it.
- **Today's topic**: **Customizing a class.**

Quote for the day:
I have traveled the length and breadth of this country and talked with the best people, and I can assure you that data processing is a fad that won't last out the year.
--Editor in charge of business books for Prentice Hall, 1957

CS100J

Reading for this lecture: Section 1.4

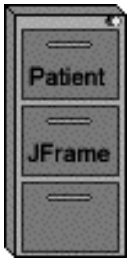
Read all the "style notes", too, and the referenced PLive lectures (activities).

Summary of lectures: On course home page, click on "Handouts" and then "Outline of lectures held so far".

Today: Your first class definition and method declaration. We will "customize" class JFrame to suit our needs.

Introduce you to Javadoc (see top of page 378).

A class is a file-drawer. Contents: manila folders.



Bill

Patient

name "B. Clinton"

address "New York"

owes $250.00

getName()

deposit(double d)

(1) unique name on tab of manila folder.

(2) manila folder, instance, object of the class

(3) fields (variables)

(4) methods (procedures and functions): instructions to do tasks

x 6 **int**

y Bill **Patient**

Bill

Patient

name "B. Clinton"

address "New York"

owes $250.00

getName()

deposit(double d)

x has value 6 y has value **Bill**

y.getName() has the value "B.Clinton"

y.deposit(250) ; will change the value of field owes to 0.

Class javax.swing.JFrame: an object is a window on your monitor.

x j1

y j2

j1

JFrame

setTitle(String) getTitle()

getX() getY() setLocation(int,int)

getWidth() getHeight() setSize(int,int)

…

new JFrame()

Expression: create a new object of class JFrame and yield its name

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)
}
```

A class definition goes in its own file named

<class-name> . java

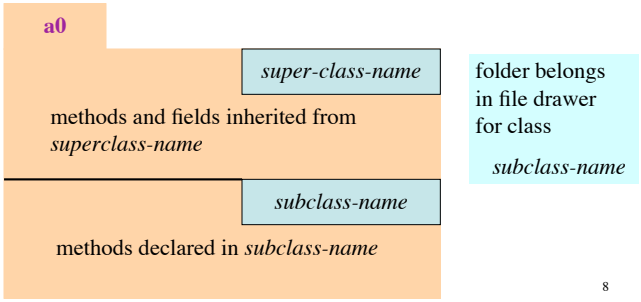
On your hard drive, have a separate directory for each Java program that you write; put all the class definitions for the program in that directory.

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 <super-class-name>   {
     declarations of methods (in any order)

}
```

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


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

        declarations of methods
}
```

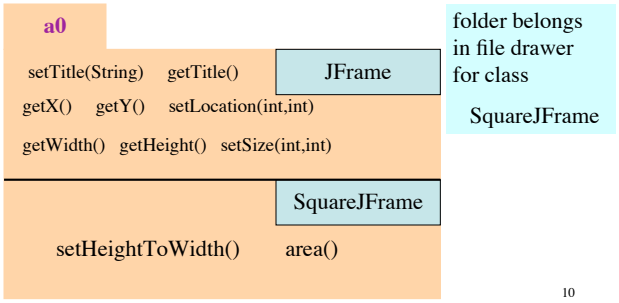


First example of a procedure and of a function

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



Javadoc

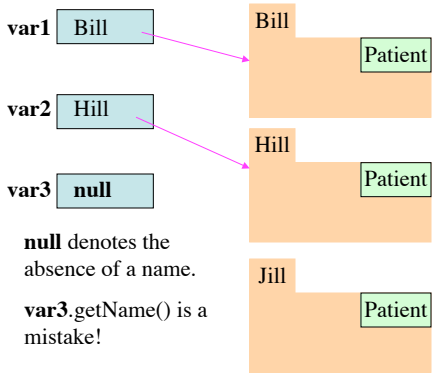
```
import javax.swing.*;
/** An instance is a JFrame with methods to square it and
    to provide the area of the JFrame */
public class SquareJFrame extends JFrame {
        /** = the area of the window */
        public int area() { … }

        /** 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.

About null



null denotes the absence of a name.

var3.getName() is a mistake!