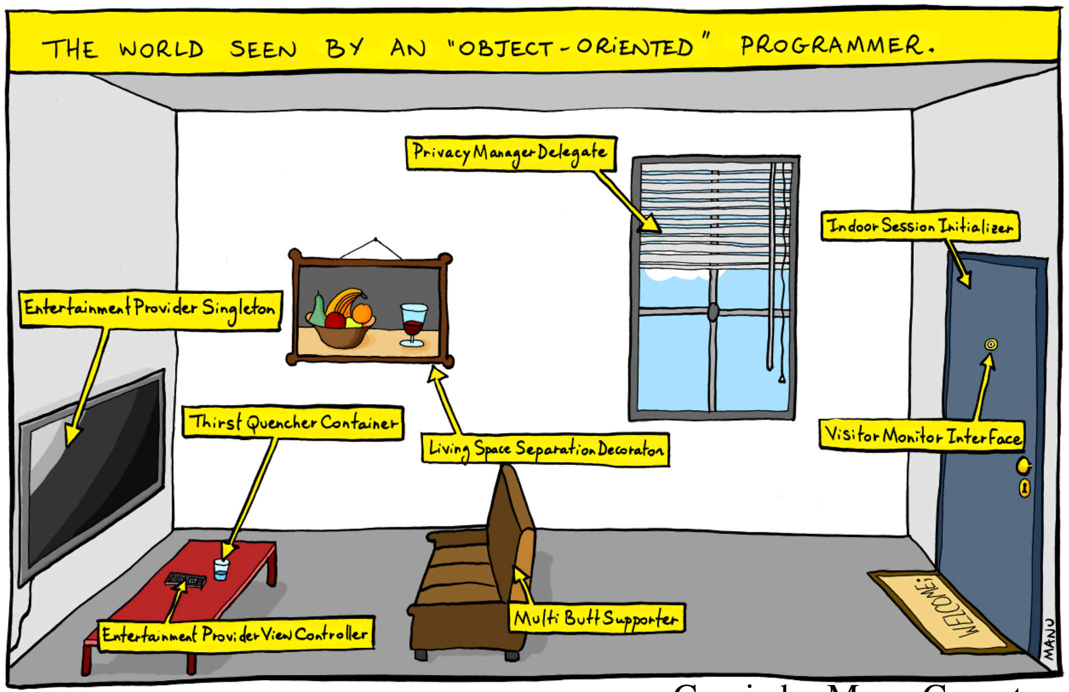
Lecture 3

Objects & The Java API

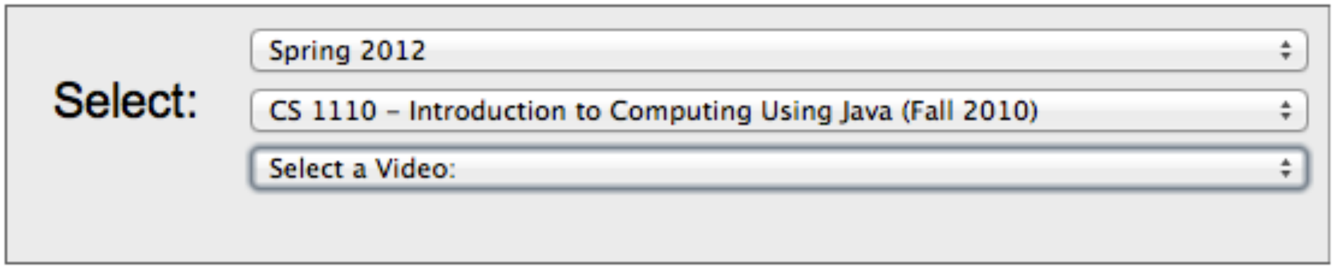
Readings for this Lecture

- Section 1.3, 1.4
  - **Study these sections**
  - **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.
- (Old) Lecture on VideoNote


THE WORLD SEEN BY AN "OBJECT-ORIENTED" PROGRAMMER.
PrivacyManagerDelegate
IndoorSessionInitializer
EntertainmentProviderSingleton
ThirstQuencherContainer
LivingSpaceSeparationDecorator
VisitorMonitorInterface
MultiButtSupporter
EntertainmentProviderViewController
WELCOME!
MANU

Comic by Manu Cornet
www.bonkersworld.net

VideoNote Lectures


Select:

Spring 2012

CS 1110 – Introduction to Computing Using Java (Fall 2010)

Select a Video:

- There are VideoNote lectures for this class
  - Recorded in Fall 2010 with David Gries
- Helpful as a resource but not a replacement
  - Style is slightly different from mine
  - Will cover topics in different orders

Quiz

- Get out a blank piece of paper.
- Write your LAST name, FIRST name, and Cornell NetID (not your Cornell ID. My NetID is wmw2)
- Put two variables on the page, exactly like this:

$$x\ \boxed{9\quad\quad}\ \textbf{int} \qquad\qquad y\ \boxed{4\quad\quad}\ \textbf{int}$$

- Write, in English, how to execute this assignment statement (but do not explain how to evaluate the expression 2*x+y–1):

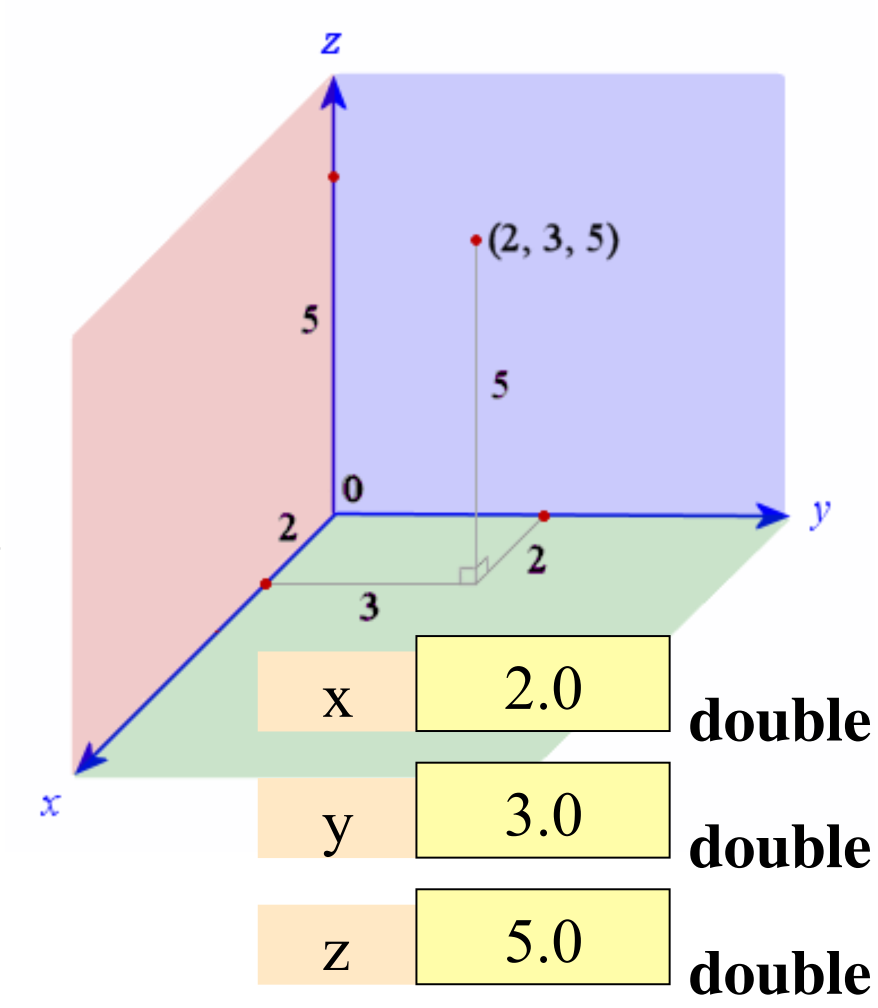
  y = 2*x + y – 1;

- Execute this assignment statement, using the two variables you previously drew on your piece of paper.

Warning: a lot of terminology today

Type: Set of values and the operations on them

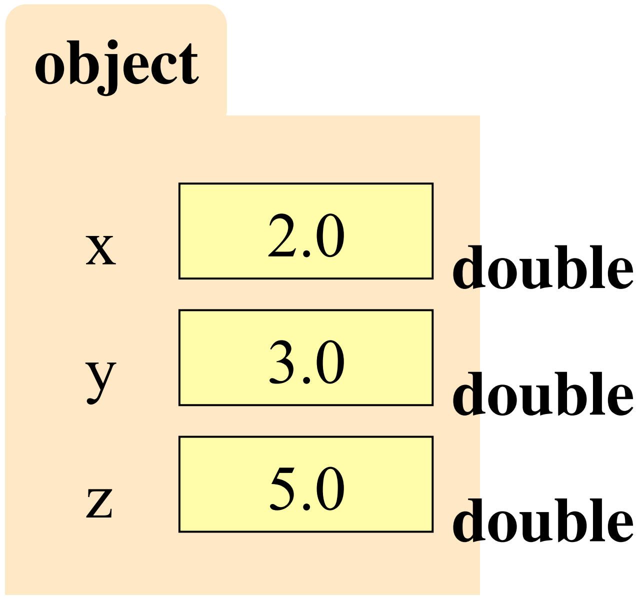
- Suppose we want to compute with a 3D point
- We need three variables
  - *x*, *y*, *z* coordinates
  - Each has type double
- What if have a lot of points?
  - Vars x0, y0, z0 for first point
  - Vars x1, y1, z1 for next point
  - …
  - This can get really messy



Type: Set of values and the operations on them

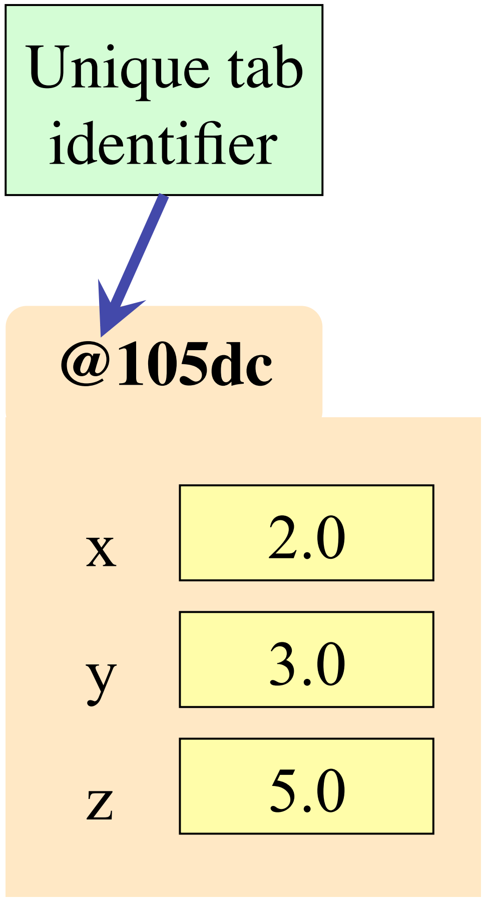
- Suppose we want to compute with a 3D point
- We need three variables
  - *x*, *y*, *z* coordinates
  - Each has type double
- What if have a lot of points?
  - Vars x0, y0, z0 for first point
  - Vars x1, y1, z1 for next point
  - …
  - This can get really messy

- Can we stick them together in a "folder"?
- This is the motivation for **objects**

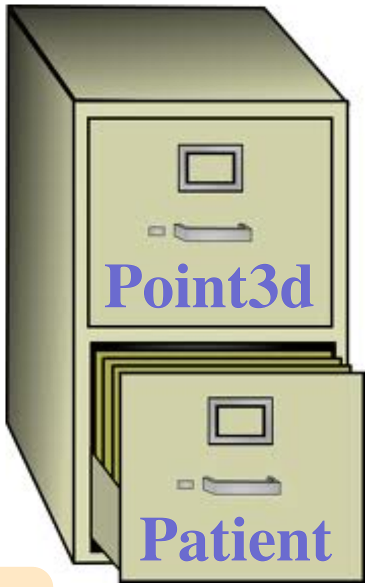


Objects: Organizing Data in Folders

- An object is like a **manila folder**
- It contains other variables
  - These variables are called **fields**
  - You can change the values of these variables (with assignments)
- It has a "tab" that identifies it
  - You cannot change this
  - Java assigns it automatically
  - More on this in demo later

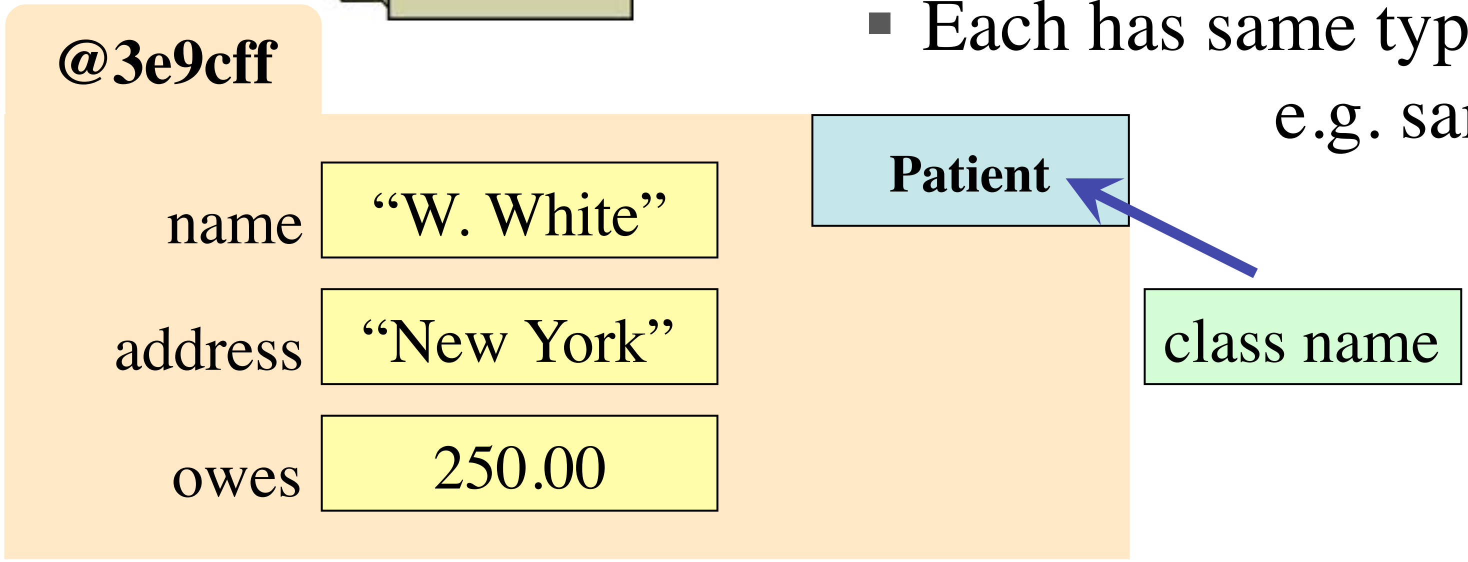


Classes: Types for Objects



- All values must have a type
  - An object type is a **class**
  - But it is more than that...
- A class is like a **file drawer**
  - Contains manila folders
  - Each has same type of info

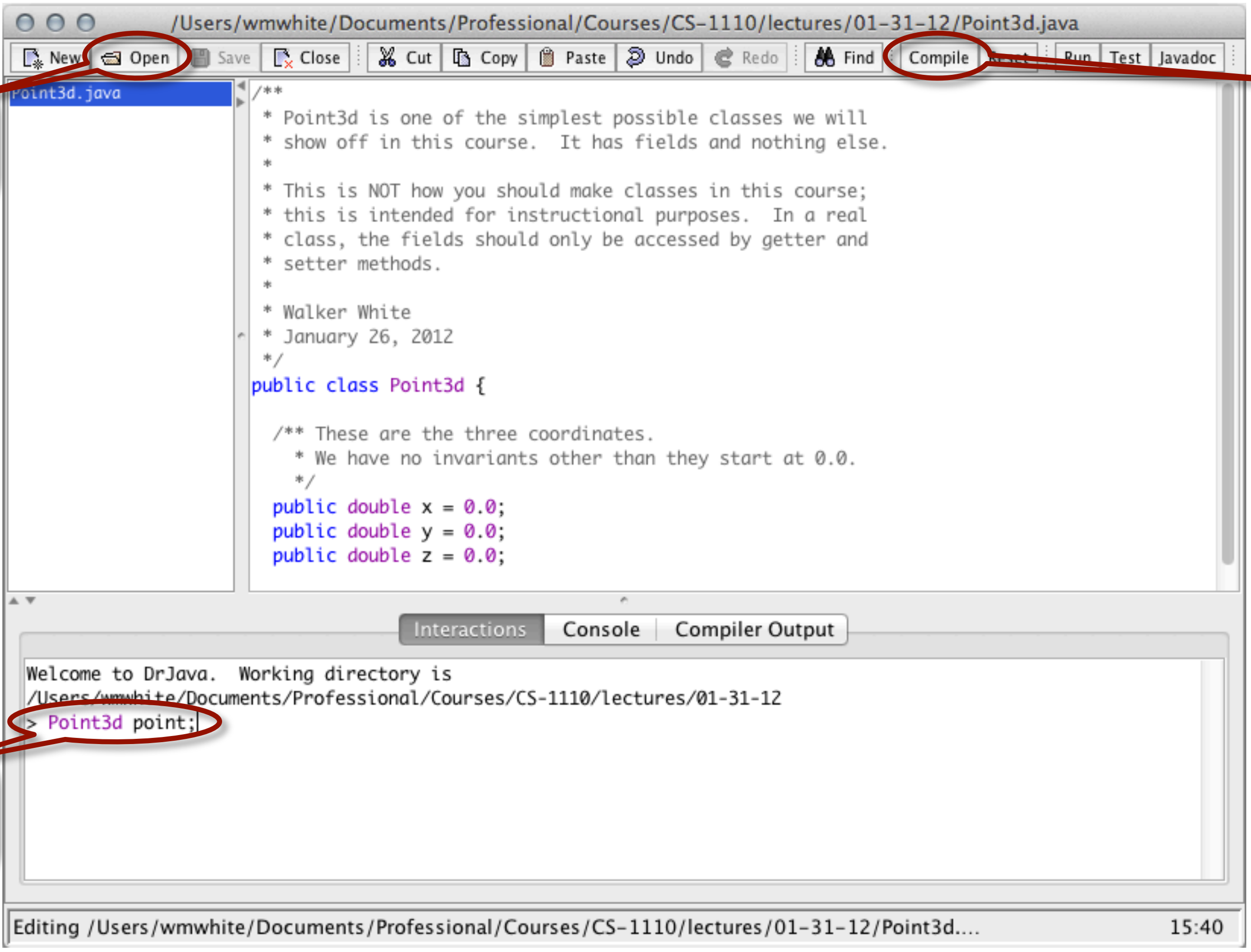
    e.g. same fields



Terminology

- **Programming language** (Java, C, Fortran, Matlab, Python): A language in which people write programs, often to be executed on a computer.
- **Program**: **All** Java programs are Classes
  A set of instructions, written in a programming language, to be executed (carried out, performed) to get some task done. Like a recipe in a cookbook.
- **Machine language**: The language of instructions that a computer is able to execute (carry out, perform).
- **Java Compiler**: Classes must be compiled to use in DrJava
  A program that translates a Java program into a machine language form so that it can be executed on a computer.

Compiling a Class


/Users/wmwhite/Documents/Professional/Courses/CS-1110/lectures/01-31-12/Point3d.java
New
Open
Save
Close
Cut
Copy
Paste
Undo
Redo
Find
Compile
Reset
Run
Test
Javadoc
Point3d.java
Interactions
Console
Compiler Output
Editing /Users/wmwhite/Documents/Professional/Courses/CS-1110/lectures/01-31-12/Point3d....
15:40

```
/**
 * Point3d is one of the simplest possible classes we will
 * show off in this course.  It has fields and nothing else.
 *
 * This is NOT how you should make classes in this course;
 * this is intended for instructional purposes.  In a real
 * class, the fields should only be accessed by getter and
 * setter methods.
 *
 * Walker White
 * January 26, 2012
 */
public class Point3d {

  /** These are the three coordinates.
    * We have no invariants other than they start at 0.0.
    */
  public double x = 0.0;
  public double y = 0.0;
  public double z = 0.0;
```

```
Welcome to DrJava.  Working directory is
/Users/wmwhite/Documents/Professional/Courses/CS-1110/lectures/01-31-12
> Point3d point;
```

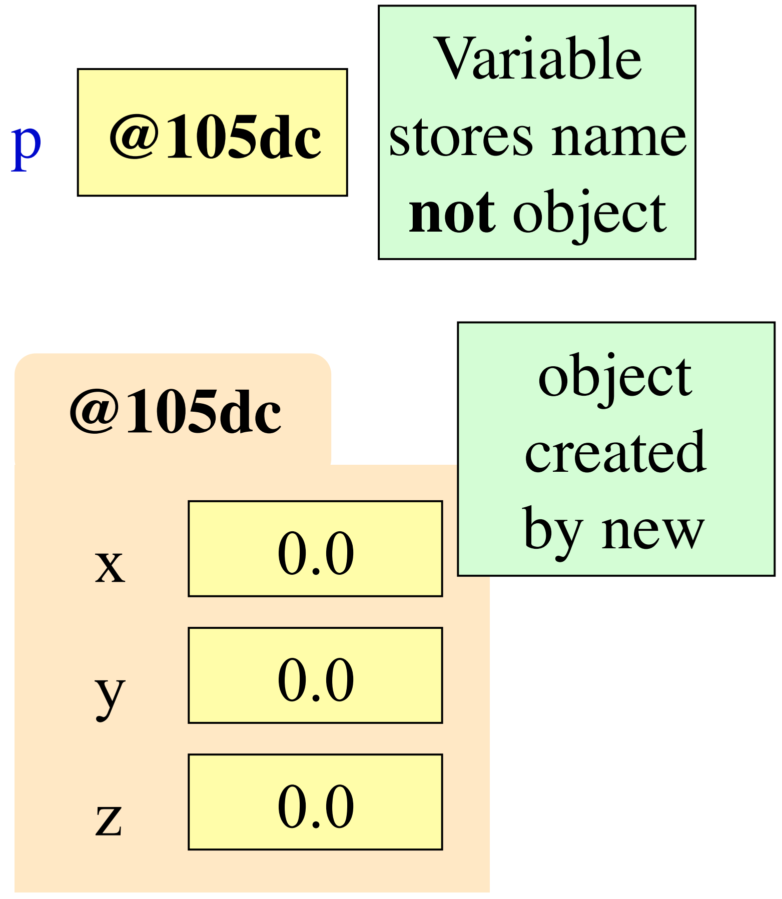
Load a .java file Point3d

Press to compile

Must compile Point3d to use

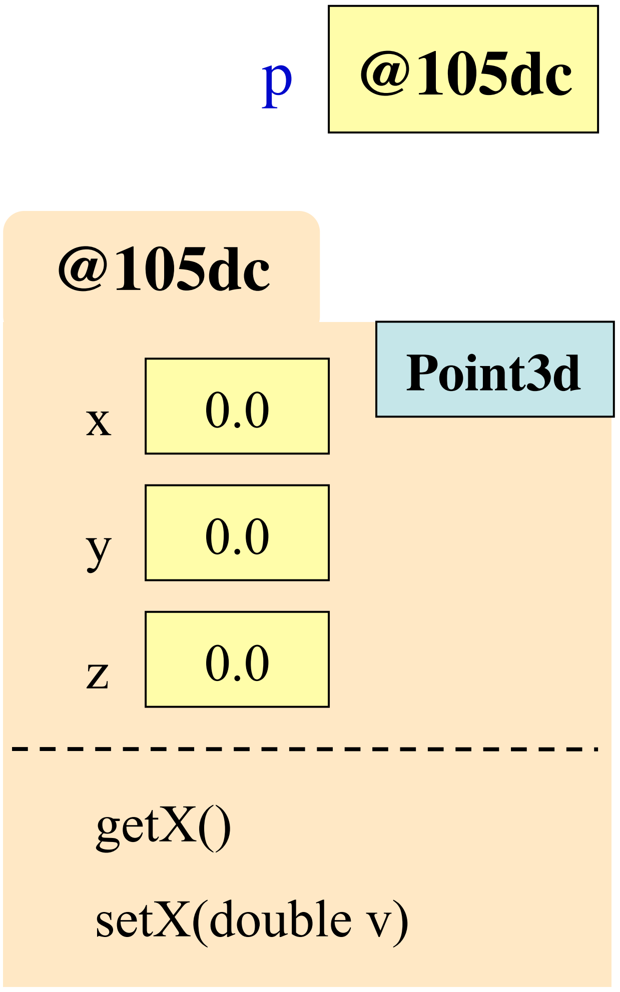
Object Initialization (the **new** keyword)

- **new** Point3d()
  - An **expression** (produces a value).
  - It creates a object (manila folder) and puts it in the file-drawer Point3d.
  - The value it gives is the tab name of the folder.
- p = **new** Point3d();
  - An **assignment statement**.
  - It computes the value of **new** Point3d() and stores this value (the tab name) in p.



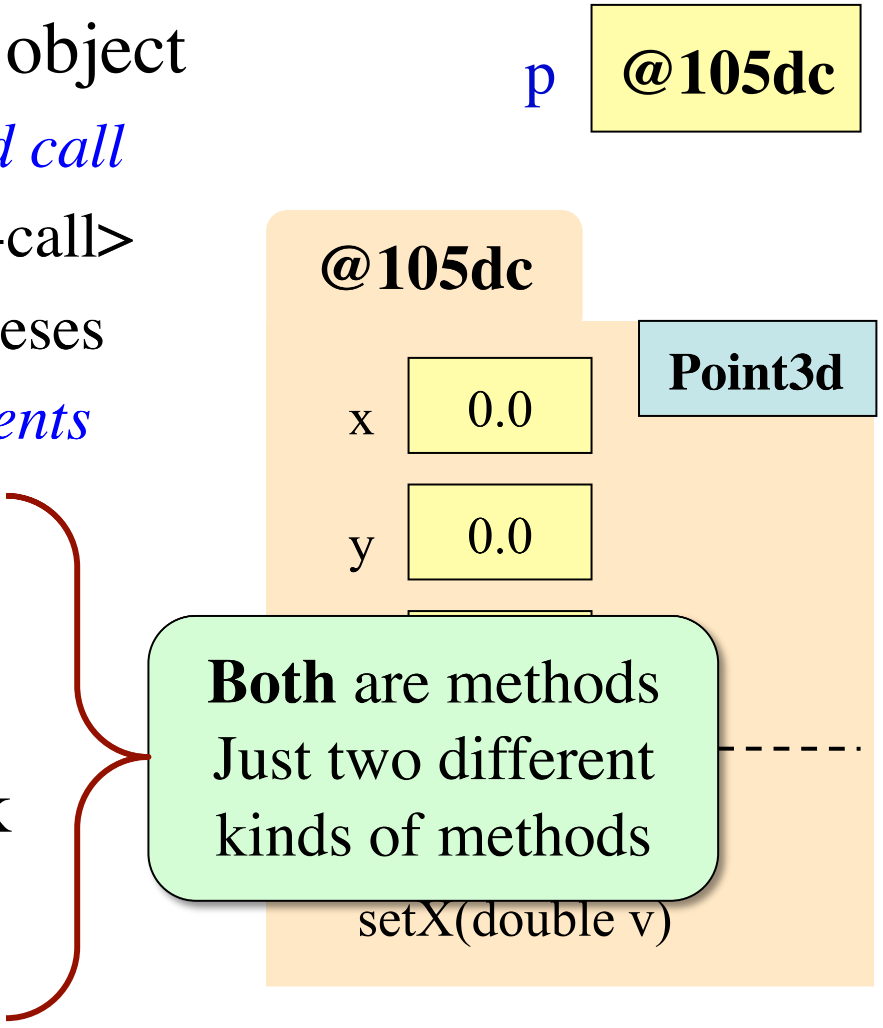
Methods: Operations on Objects

- Method: instruction for an object
  - Use of a method is a *method call*
  - <object-variable>.<method-call>
  - Method calls end in parentheses
  - Values in parens are *arguments*
- **Functions** return a value
  - Works like an *expression*
  - **Example**: p.getX()
- **Procedures** perform a task
  - Works like a *command*
  - **Example**: p.setX(3.4);



Methods: Operations on Objects

- Method: instruction for an object
  - Use of a method is a *method call*
  - <object-variable>.<method-call>
  - Method calls end in parentheses
  - Values in parens are *arguments*
- **Functions** return a value
  - Works like an *expression*
  - **Example**: p.getX()
- **Procedures** perform a task
  - Works like a *command*
  - **Example**: p.setX(3.4);



Both are methods
Just two different
kinds of methods

Exercise: Objects and Assignment

- What is the value of q?

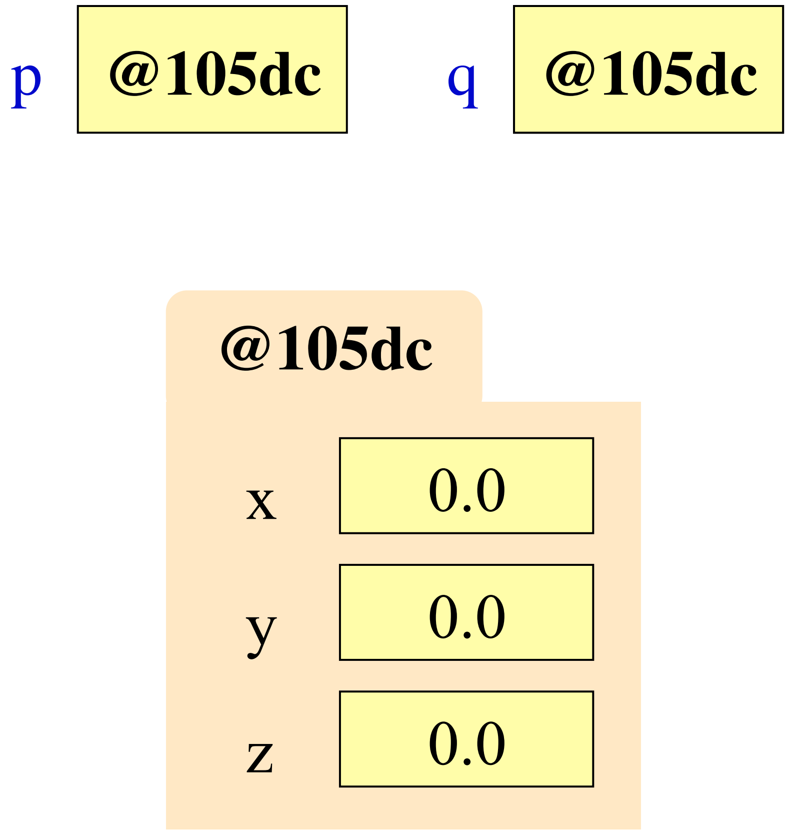
```
Point3d p = new Point3d();
Point3d q = p;
```

- Execute the commands:

```
p.setX(5.6);
q.setX(7.4);
```

- What is value of p.getX()?

A: 5.6
B: 7.4 **CORRECT**
C: @105dc
D: I don't know

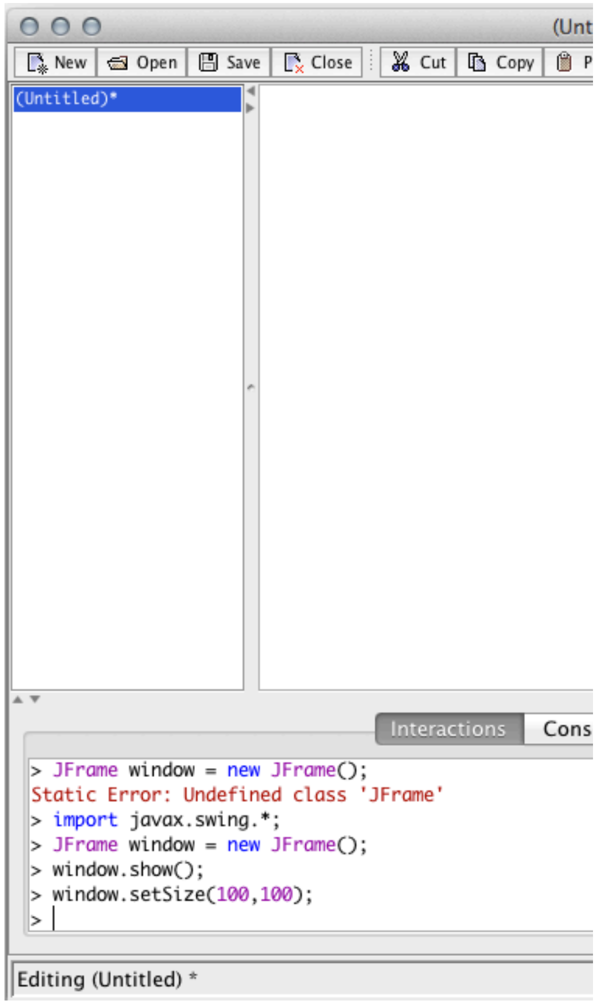


Comments from Previous Semesters

- 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.

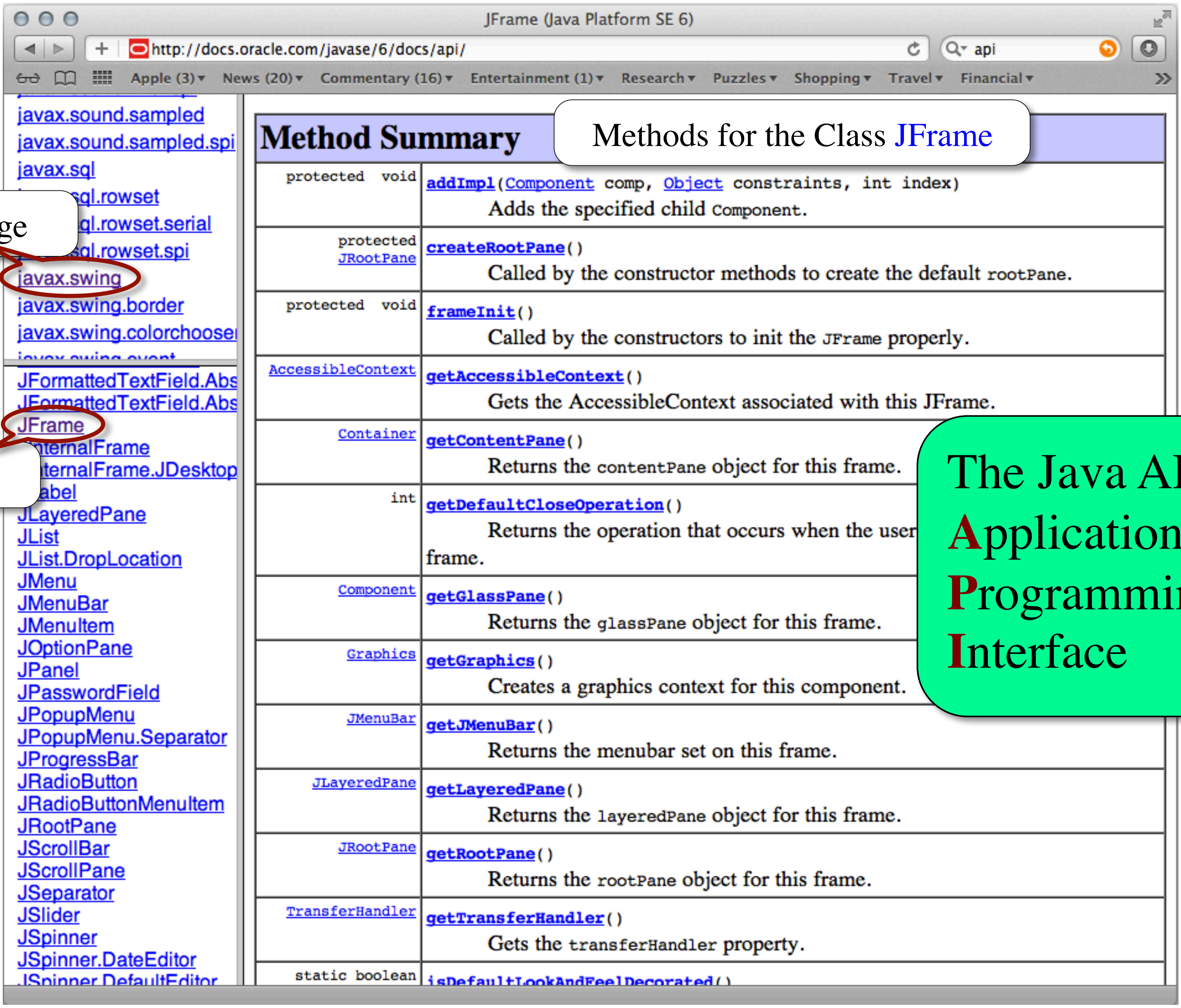
Packages and Built-in Classes

- Java has built-in classes
  - No need to compile them
  - But you have to import them
- Built-in classes are in **packages**
  - Use a command to import
  - **import <package>.<class>;**
  - **import <package>.*;**
    - imports everything in package
- **Example**: JFrame
  - Java class for (empty) Window
  - In package javax.swing



```
> JFrame window = new JFrame();
Static Error: Undefined class 'JFrame'
> import javax.swing.*;
> JFrame window = new JFrame();
> window.show();
> window.setSize(100,100);
> |
```

Editing (Untitled) *


JFrame (Java Platform SE 6)
http://docs.oracle.com/javase/6/docs/api/
api
Apple (3) News (20) Commentary (16) Entertainment (1) Research Puzzles Shopping Travel Financial

javax.sound.sampled
javax.sound.sampled.spi
javax.sql
javax.sql.rowset
javax.sql.rowset.serial
javax.sql.rowset.spi
javax.swing
javax.swing.border
javax.swing.colorchooser

JFormattedTextField.Abs
JFormattedTextField.Abs
JFrame
JInternalFrame
JInternalFrame.JDesktop
JLabel
JLayeredPane
JList
JList.DropLocation
JMenu
JMenuBar
JMenuItem
JOptionPane
JPanel
JPasswordField
JPopupMenu
JPopupMenu.Separator
JProgressBar
JRadioButton
JRadioButtonMenuItem
JRootPane
JScrollBar
JScrollPane
JSeparator
JSlider
JSpinner
JSpinner.DateEditor
JSpinner.DefaultEditor

Package

Class

Method Summary

Methods for the Class JFrame

protected void	`addImpl(Component comp, Object constraints, int index)` Adds the specified child `Component`.
protected JRootPane	`createRootPane()` Called by the constructor methods to create the default `rootPane`.
protected void	`frameInit()` Called by the constructors to init the `JFrame` properly.
AccessibleContext	`getAccessibleContext()` Gets the AccessibleContext associated with this JFrame.
Container	`getContentPane()` Returns the `contentPane` object for this frame.
int	`getDefaultCloseOperation()` Returns the operation that occurs when the user frame.
Component	`getGlassPane()` Returns the `glassPane` object for this frame.
Graphics	`getGraphics()` Creates a graphics context for this component.
JMenuBar	`getJMenuBar()` Returns the menubar set on this frame.
JLayeredPane	`getLayeredPane()` Returns the `layeredPane` object for this frame.
JRootPane	`getRootPane()` Returns the `rootPane` object for this frame.
TransferHandler	`getTransferHandler()` Gets the `transferHandler` property.
static boolean	`isDefaultLookAndFeelDecorated()`

The Java API:
Application
Programming
Interface

String is a Class!

String s = “Hello World”;

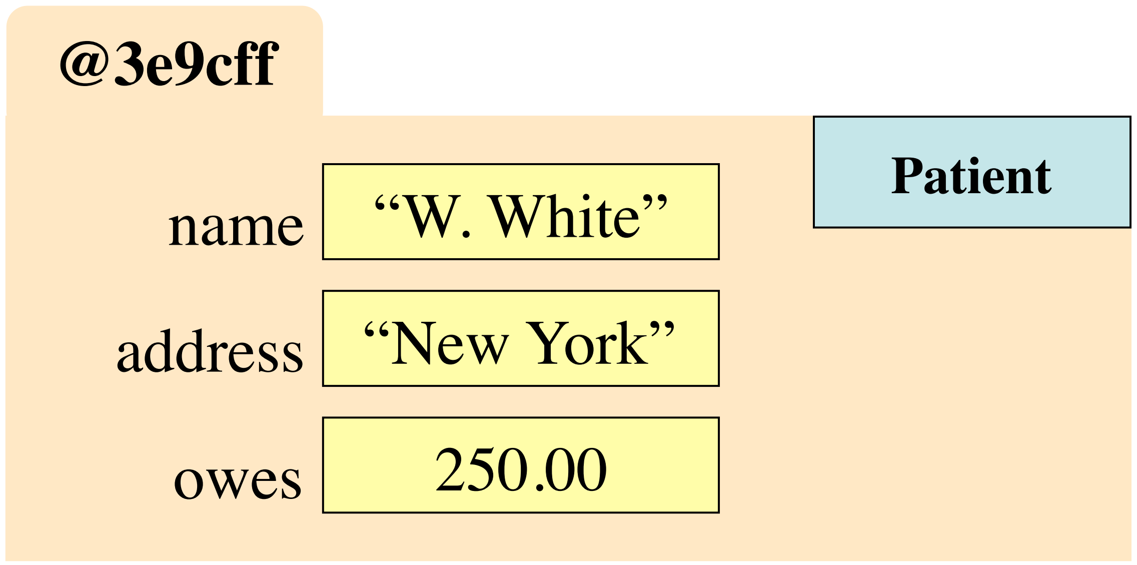
- Different from other classes
  - Do **not** create with new
- In package java.lang
  - Imported by default
  - Never need to import
- Great class to “play with”
  - All methods are functions
  - Use in interactions pane

String Methods

- **charAt(int p)**
  Get letter at position p
- **substring(int p)**
  Get suffix starting at position p
- **substring(int p, int e)**
  Get suffix starting at postion p, ending at e-1

Where To From Here?

- OO Programming is about **creating classes**
  - You will learn to make your own classes
  - You will learn what you can do with methods
- Understanding classes and objects is important


@3e9cff
Patient
name
"W. White"
address
"New York"
owes
250.00