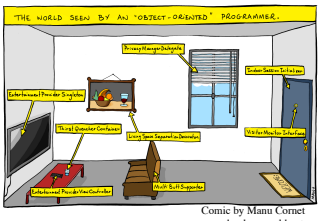
Readings for this Lecture

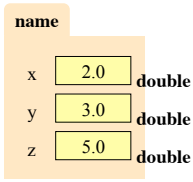
- Section 1.3, 1.4
  - **Study these sections**
  - **Practice** what is taught using DrJava.
- **PLive**: Activities 3-3.1, 3-3.2, 3-3.4 (not 3-3.3), 3-4.1, 3-4.2.
- (Old) Lecture on VideoNote



Comic by Manu Cornet
www.bonkersworld.net

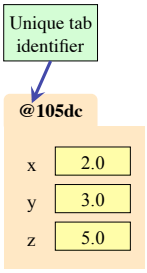
Type: Set of values and the operations on them

- 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**

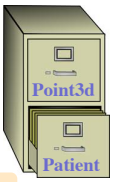


Objects: Organizing Data in Folders

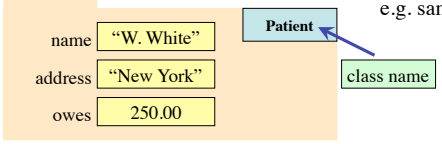
- An object is like a **manila folder**
- It contains other variables
  - These variables are called **fields**
  - You can change the values of these variables (with assignments)
- It has a "tab" that identifies it
  - You cannot change this
  - Java assigns it automatically
  - More on this in demo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



Classes: Types for Objects



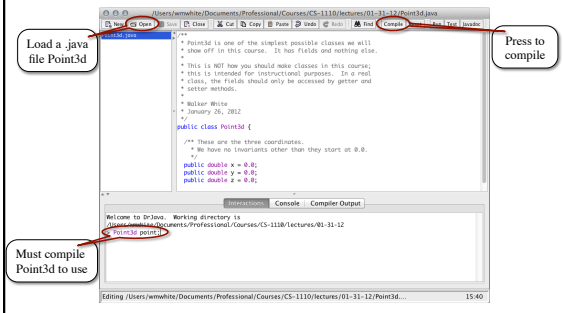
- All values must have a type
  - An object type is a **class**
  - But it is more than that...
- A class is like a **file drawer**
  - Contains manila folders
  - Each has same type of info e.g. same fields



Terminology

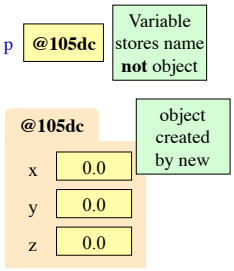
- **Programming language** (Java, C, Fortran, Matlab, Python): A language in which people write programs, often to be executed on a computer.
- **Program**: **All** Java programs are Classes
  A set of instructions, written in a programming language, to be executed (carried out, performed) to get some 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



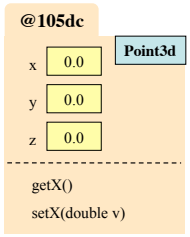
Object Initialization (the new keyword)

- **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.



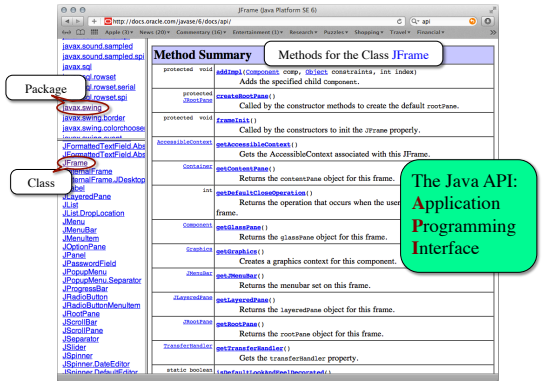
Methods: Operations on Objects

- 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);



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



String is a Class!

String s = "Hello World";

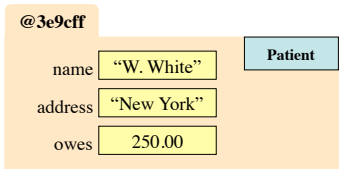
- Different from other classes
  - Do **not** create with new
- In package java.lang
  - Imported by default
  - Never need to import
- Great class to "play with"
  - All methods are functions
  - Use in interactions pane

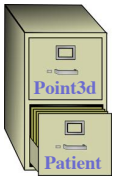
String Methods

- **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





1/26/12