What's in Compsci 100?

- Understanding tradeoffs: reasoning, analyzing, describing…
  - Algorithms
  - Data Structures
  - Programming
  - Design
- Object oriented programming using Java
  - IDE
  - Language
  - Problem-solving
  - From design to code

Toward understanding data structures

- What can be put in a TreeSet?
- What can be sorted?
  - Where do we find this information?
  - How do we understand the information?

- What can be put in an ArrayList? Why is this different?
  - What operations exist on an ArrayList?
  - What about an array, or operations done on an ArrayList as opposed to what an ArrayList does to itself?

What can an Object do (to itself)?

- http://www.cs.duke.edu/csed/java/jdk1.5/docs/api/index.html
  - Look at java.lang.Object
  - What is this class? What is its purpose?

- `toString()`
  - Used to print (`System.out.println`) an object
  - overriding `toString()` useful in new classes
  - String concatenation: `String s = "value "+x;`
  - Default is basically a pointer-value

What else can you do to an Object?

- `equals(Object o)`
  - Determines if guts of two objects are the same, must override, e.g., for using `a.indexOf(o)` in ArrayList a
  - Default is ==, pointer equality

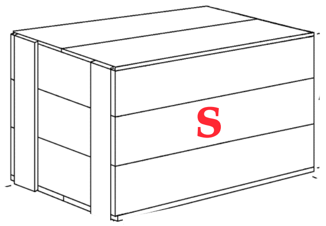
- `hashCode()`
  - Hashes object (guts) to value for efficient lookup

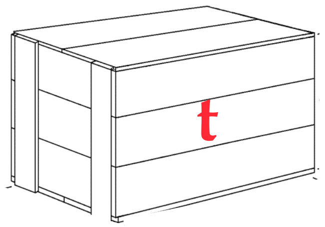
- If you're implementing a new class, to play nice with others you *must*
  - Override `equals` and `hashCode`
  - Ensure that equal objects return same `hashCode` value

Objects and values

- Primitive variables are boxes
  - think memory location with value
- Object variables are labels that are put on boxes

```
String s = new String("genome");
String t = new String("genome");
if (s == t) {they label the same box}
if (s.equals(t)) {contents of boxes the same}
```





What's in the boxes? "genome" is in the boxes

Objects, values, classes

- For primitive types: int, char, double, boolean
  - Variables have names and are themselves boxes (metaphorically)
  - Two int variables assigned 17 are equal with ==

- For object types: String, ArrayList, others
  - Variables have names and are labels for boxes
  - If no box assigned, created, then label applied to ***`null`***
  - Can assign label to existing box (via another label)
  - Can create new box using built-in ***`new`***

- Object types are references or pointers or labels to storage

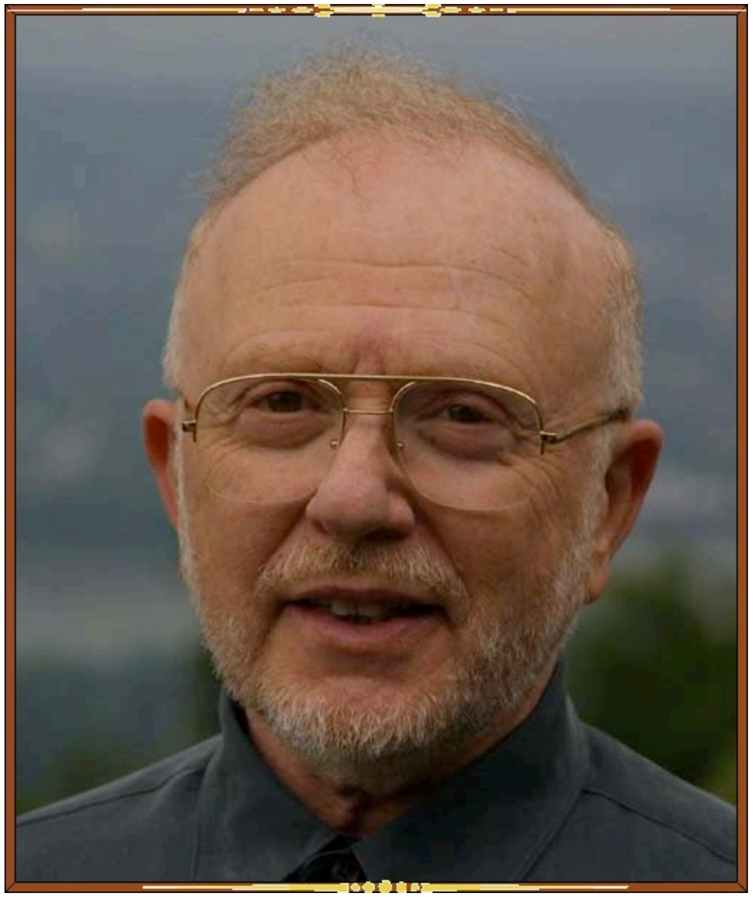
Anatomy of a class

```
public class Foo {
     private int mySize;
     private String myName;
     public Foo(){
          // what's needed?
     }
     public int getSize(){
          return mySize;
     }
     public double getArea(){
          double x;
          x = Math.sqrt(mySize);
          return x;
     }
}
```

- What values for vars (variables) and ivars (instance variables)?

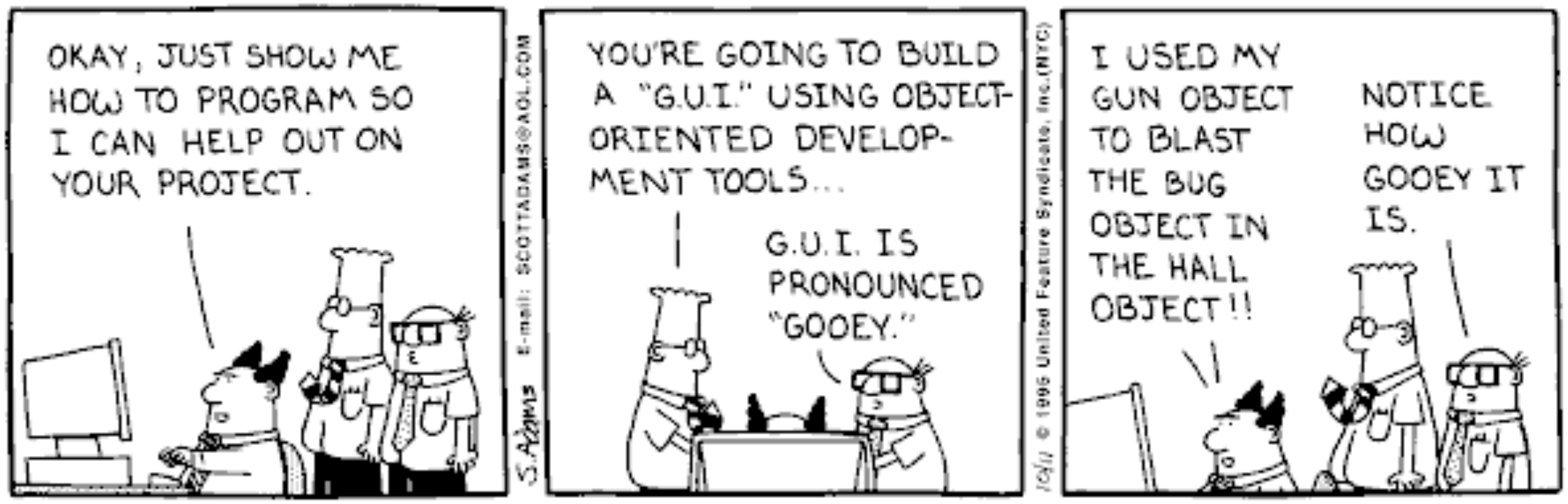
David Parnas

"For much of my life, I have been a software voyeur, peeking furtively at other people's dirty code. Occasionally, I find a real jewel, a well-structured program written in a consistent style, free of kludges, developed so that each component is simple and organized, and designed so that the product is easy to change. "



Parnas on re-invention

"We must not forget that the wheel is reinvented so often because it is a very good idea; I've learned to worry more about the soundness of ideas that were invented only once. "


OKAY, JUST SHOW ME HOW TO PROGRAM SO I CAN HELP OUT ON YOUR PROJECT.
YOU'RE GOING TO BUILD A "G.U.I." USING OBJECT-ORIENTED DEVELOPMENT TOOLS...
G.U.I. IS PRONOUNCED "GOOEY."
I USED MY GUN OBJECT TO BLAST THE BUG OBJECT IN THE HALL OBJECT!!
NOTICE HOW GOOEY IT IS.
S. Adams E-mail: SCOTTADAMS@AOL.COM
10/11 © 1995 United Feature Syndicate, Inc.(NYC)

David Parnas (entry in Wikipedia)

- *Module Design:* Parnas wrote about the criteria for designing modules, in other words, the criteria for grouping functions together. This was a key predecessor to designing objects, and today's object-oriented design.
- *Social Responsibility:* Parnas also took a key stand against the Strategic Defense Initiative (SDI) in the mid 1980s, arguing that it would be impossible to write an application that was free enough from errors to be safely deployed.
- *Professionalism:* Parnas became one of the first software engineers to earn a professional engineering license in Canada. He believes that software engineering is a branch of traditional engineering.

What about a 'struct' (plain old data)

- We use classes, data/state is private, methods are public
  - Why do we have rules? When can they be broken?
  - Why are there both structs and classes in C++?

- What about helping class, e.g., word and frequency together?
  - We can have one class nested in another, then we don't have to worry so much about *encapsulation*

- We'll see example for a new class that can be compared using equality and can be sorted
  - Comparable interface must be symmetric with .equals
  - What happens if this isn't the case? Sometimes ok?

John von Neumann

“Anyone who attempts to generate random numbers by deterministic means is, of course, living in a state of sin.”

“There's no sense in being precise when you don't even know what you're talking about. “

“There are two kinds of people in the world: Johnny von Neumann and the rest of us.”

Eugene Wigner, Noble Physicist

Tomato and Tomato, how to code

- `java.util.Collection` and `java.util.Collections`
  - one is an interface
    - `add(), addAll(), remove(), removeAll(), clear()`
    - `toArray(), size(), iterator()`
  - one is a collection of static methods
    - `sort(), shuffle(), reverse(), max()`
    - `frequency(), indexOfSubList ()`
- `java.util.Arrays`
  - Also a collection of static methods
    - `sort(), fill(), binarySearch(), asList()`