

# Classes and Inheritance

# An Interesting Project

Folding@home

<http://folding.stanford.edu/>

# Announcements

- Breakout due right now.
- Assignment 4 (Hangman) out, due Wednesday, February 22 at 3:15PM.
  - Play around with strings, graphics, and file processing!
  - Sharpen your vocabulary!
- Midterm review session Sunday, 7-8:30PM in Hewlett 200.
- Friday Four Square!
  - Today at 4:15 outside Gates.

# Creating our own Class



# Constructors

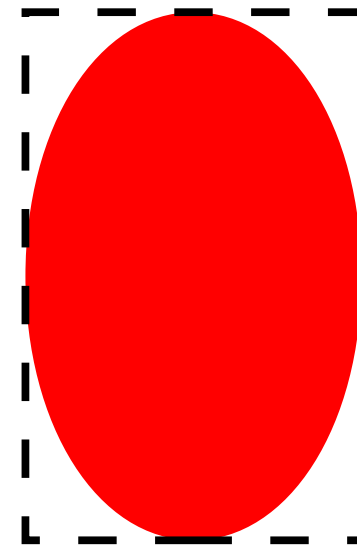
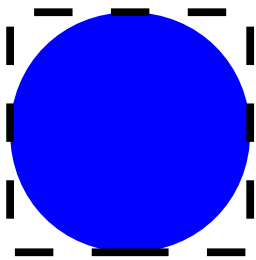
- A **constructor** is a special method defined in a class that is responsible for setting up class's instance variables to appropriate values.
- Syntax:

```
public NameOfClass (parameters) {  
    /* ... body of constructor ... */  
}
```

- Inside a constructor:
  - Give initial values to instance variables.
  - Set up instance variables based on values specified in the parameters.
- Constructor called when instance created with **new**.

# Instance Variables Revisited

- Each instance of a class gets its own, unique copy of each instance variable.
- Different instances of the same object cannot read or write each others' instance variables.



# toString()

- To get a string representation of an object, Java uses a method

```
public String toString()
```

- If you define this method in your Java classes, you can customize what string will be produced.
- Otherwise, you get Icky Javaspeak string representations.



**ONE DOES NOT SIMPLY**

**ACCESS PRIVATE IVARS**

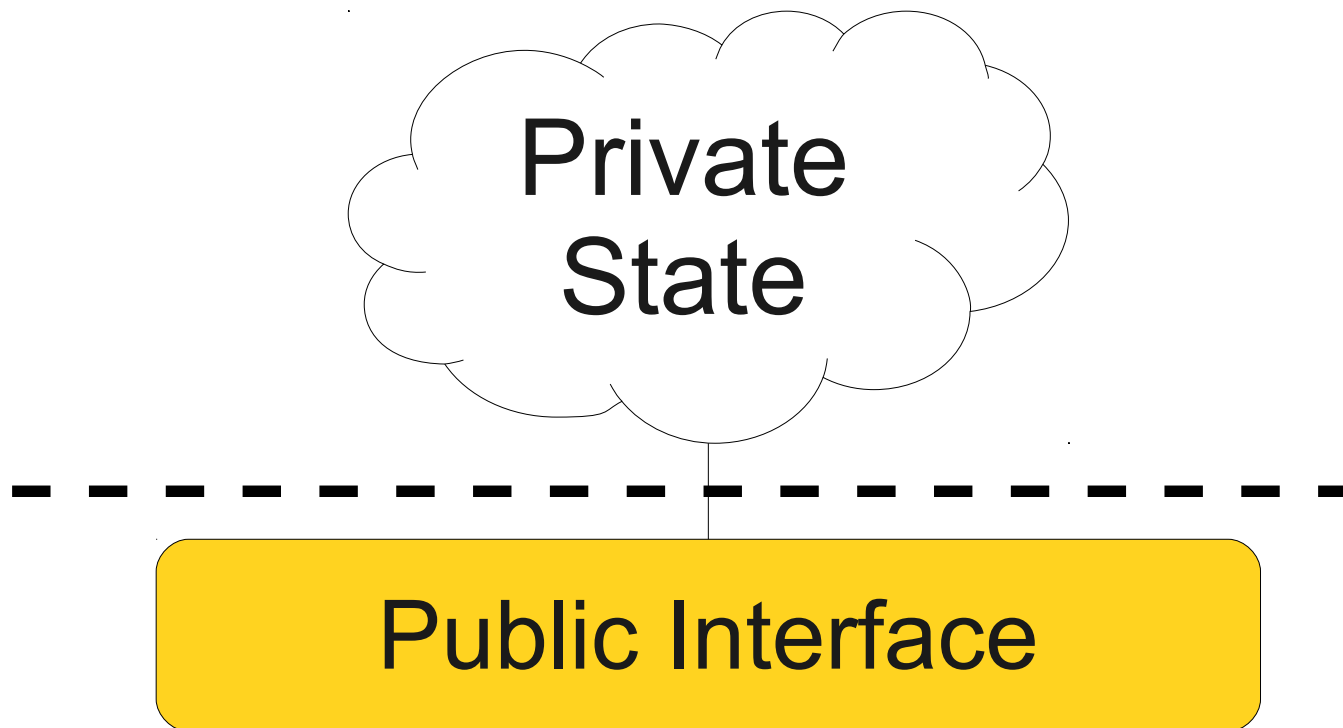


# public and private

- A method or instance variable declared **public** can be accessed from *anywhere*.
- A method or instance variable declared **private** can only be accessed by an instance of the class.

# public and private

- A method or instance variable declared **public** can be accessed from *anywhere*.
- A method or instance variable declared **private** can only be accessed by an instance of the class.



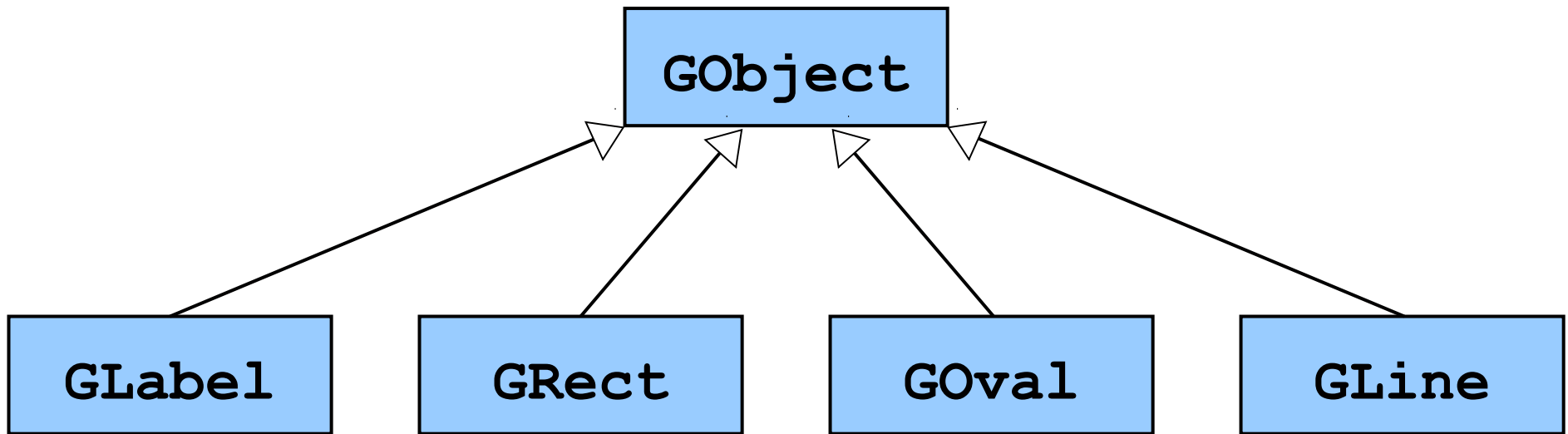
# Inheritance

- A class can be defined as a refinement of some other class.
- The new class **inherits** from that class.
- The original class is the **superclass**; the new class is a **subclass**.
- Syntax:

```
public class Name extends Superclass {  
    /* ... */  
}
```

# The GObject Hierarchy

The classes that represent graphical objects form a hierarchy, part of which looks like this:



# Operations on GObject

```
void setColor(Color c)
```

```
void setLocation(double x, double y)
```

```
void move(double dx, double dy)
```

```
double getX()
```

```
double getY()
```

```
double getWidth()
```

```
double getHeight()
```