COMS W1114 - Java Lab

Lab 11
Wednesday, April 14, 2004
&
Thursday, April 15, 2004

-1-

Notes

- HW5 out. Due Tuesday 11a. Any questions?
- Only three more labs :(
  - Today: GUI programming! awt and Swing (Ch 9)
  - April 22 - Event-based programming (Ch 10)
  - April 29 - 1) applets and 2) code packaging/APIs
- Homework 6 out next Tuesday-ish

-2-

Lab 10 Review (1)

- this identifier - The use of this is only needed when there is ambiguity over variable names in a particular scope.
- overloading a method - where you provide different versions of a method, but keep the same name.
- overriding methods – a subclass can decide to supply its own version of a method already supplied.
- Object class methods - ie. toString(), equals, etc.
- instanceOf operator - tests whether its first operand is an instance of its second.
  - boolean val = op1 instanceof op2;
  - op1 must be the name of an object and op2 must be the name of a class. An object is considered to be an instance of a class if that object directly or indirectly descends from that class.

-3-

Lab 10 Review (2)

- Inheritance - An object is considered to be an instance of a class if that object directly or indirectly descends from that class.
- Protected - a protected variable can be accessed ONLY from this class and the classes descend from this class.
- Static
  - a static variable - a member (global) variable which exists only ONCE even though there may be multiple objects created.
  - a static method - not part of the a specific object, part of the general class

-4-

Graphics and UIs

- Two packages for dealing with graphics (for now)
  - java.awt (awt - the abstract windowing toolkit)
  - javax.swing (swing)
- Each provides access to tools/code for writing GUIs, drawing, etc.
- awt uses much of the OS's facilities - so UIs look like the platform they are run on
- swing is implemented independently of the OS
- They each are quite large packages and, like many things in the class, you can take an entire course on them alone.
- We will start with awt, then migrate to swing next lab. They are rather similar.
- Our goal: be able to write some simple graphics programs.

-5-

Graphics and UIs

- Structure of AWT (diagram in book. pp 385)
  - graphics
  - components (windows and menus too)
  - layout managers
  - event handlers
  - image manipulation

-6-

Example 1

- Look at warning box example from the book (pp 389). We want to display a window with some text in it.
- Frame - the basic window
  - Frame is a subclass of the Window component
  - Our code will inherit the Frame code
- Add Graphics (paint and repaint methods)
- Viola!
- see sample code “Warning.java”
- Pretty simple, right?

-7-

Example 2

- Remember our old Point class from lab 9? (NOT the point class you are building for HW5!)
- Recall: modeled a point in 2D Cartesian space.
- [See javadoc for Point]
- Now, say we want to plot the points in a graph on your machine? Let’s build a Plotter2 class that is a real plotter!
- Where to start? Just like before:
  - Frame - the basic window
  - Add paint (and repaint) using the Graphics object
  - Here, our painting is a bit more imvolved.

-8-

More Graphics

- Now, what if you want more than one window or more control?
  - a Canvas
- Walkthrough book example 10.3 (FlagMaker2)

-9-

A few more graphic objects

- Frame and Canvas are great for simple drawing. What if you want to make an interactive application?
- Want Labels
  - `add(new Label ("some text"));`
- Want Buttons
  - a little more involved, but rather straightforward
  1. create a Button object

     `Button myButton = new Button("Submit");`
  2. add it to the Frame/Canvas - recall, these are Container objects. Note that Containers have this *add* method (seen with Labels)

     `add(myButton);`
- Why no x/y coordinates for the Button???
  - there is a *Layout Manager* to coordinate placement (nice :)

Layout Manager

- Layout Manger take control of the over the positioning of components and arrange them sensibly.
- There are 5 different managers! We'll only talk about three:
  - FlowLayout, BorderLayout(default) and GridLayout

```
setLayout(new Manager(parameter)); //format
```

example:

```
setLayout(new FlowLayout(FlowLayout.CENTER,horigap,vertigap));
```

We'll see it used in a minute….

Simple Event

- Make a button do something
- We have our button myButton and we've added it

  ```
  Button myButton = new Button("Submit");
    add(myButton);
  ```
- Now need to "listen" for actions/events we care about

  ```
  myButton.addActionListener (this);
  ```

  this means the current frame will be responsible for the code for some *ActionPerformed* method(what?! pretty easy….)

```
public void actionPerformed (ActionEvent e){
    if (e.getSource() == buttonname1) {
            statements;
    } else
    if (e.getSource() == buttonname2) {
            statements;
    } //etc
}
```

Putting it all together

- (See the ButtonTest code example)