The UNIVERSITY of VERMONT

***Benjamin Fenster***
CS-100 (Damon)
8 February 2003

# Notes – User Interfaces

I. Simple Windows / Basics
   a. User interface is nothing more than a collection of objects.
   b. Each UI piece is called a component.
   c. As many components as you want can be created, but they need code in the background to actually work.

II. Events
   a. When a button is clicked, the system creates an event (ActionEvent)
   b. Event looks for a listener
      i. May be part of the button itself.
      ii. May be associated with the button.
   c. An appropriate method of the listener is called.
   d. Events are designed to communicate information, so they may be practical outside of user interfaces too.
   e. May kinds of events
      i. All user interface events are subclasses of AWTEvent
      ii. Forty or fifty kinds of events exist already, plus more can be created of course.

III. Listeners
   a. In CS100 a button will always be associated with a listener, but in reality about half the time the button itself will be made a listener.
   b. Different kinds of listeners
      i. Base is EventListener
      ii. A listener exists for every existing type of event.
      iii. To actually use the existing listeners, derive a subclass.
   c. ActionListener has justone method (actionPerformed()) and an empty constructor.
   d. Other listeners have many more methods.
   e. If you don't re-implement a method, it just does nothing.

IV. Containers
   a. Can't just have a button floating free on the screen.
   b. Everything needs a container.
   c. The only top-level components are JFrame,JDialog, and JWindow
   d. Insert components in these
   e. (See slide 7Feb-11 for code sample)

V. Layout
   a. Where does each piece go? How big is it? Et cetera.
   b. Size
      i. All components have a minimum and default size.
      ii. System or components themselves may resize components.
      iii. `setPreferredSize(width, height)`
      iv. Can also set minimum and maximum size explicitly, but setPreferredSize() is usually the right tool to pick.
      v. setPreferredSize() normally sets the minimum size to be used, though in extreme cases it may be reduced. (It's hard to get it to do that)
   c. Layout Managers
      i. Simple way to do layout
      ii. Flow Layout
         1. Default. Simple.
         2. `setLayout(new FlowLayout())`
         3. Lays out components left to right, top to bottom, like words on a page.
         4. Components aren't resized.
         5. Can align { LEFT, CENTER, RIGHT }
         6. Can set minimum spacing, both vertical and horizontal.

7. Ultimately simple layouts can be created very nicely.

iii. Border Layout
   1. Works with compass points (North, South, East, West, Center)
   2. The point to use is the second argument in the `.add()` call
   3. Components will be resized depending on where they are on the screen.
      a. East and West components get resized vertically
      b. North and South components get resized horizontally.
      c. Center components get resized in both directions.
   4. Call `.pack()` when you're done with setting up layout – this finishes packing everything together.
   5. It's still fairly basic, but it's okay for simple stuff.

iv. More Complex Managers
   1. It's very rare to see simple layouts like those.
   2. Try GridLayout – make a grid and say which grid square you want.
   3. Use sub containers, and combinations of simple layouts.
      a. Use a border layout to setup three panels.
      b. Within each panel, use a flow layout to create several components.
      c. Build up complex designs that way.
   4. You can also hardwire more complex locations, etc.
      a. More about this next week.
      b. Be cautious!