2110: Announcements
Grades for Prelim 1 are now available. You should have received an email from Gradescope about accessing them.
Regrade requests will be open starting tomorrow morning.
Lunch with Professors. People have forgotten about this. There’s lots of room, today and later dates.
TODAY is GIVING DAY

2110: Next week’s recitation
Making a class that maintains a collection of values “iterable”
Interfaces Iterator and Iterable. Watch about 15 minutes of videos beforehand.

```java
int[] b = ...;
...
// Print all elements b[i]
for (int e : b) {
    System.out.println(e);
}
...
// Print the elements in the set
for (Integer e : hs) {
    System.out.println(e);
}
...
for (String e : dl) {
    System.out.println(e);
}
```

GUI (Graphical User Interface)
- Provides a friendly interface between user and program
- Allows event-driven or reactive programming: The program reacts to events such as button clicks, mouse movement, keyboard input
- Often is multi-threaded: Different threads of execution can be executing simultaneously. We study concurrency and threads in April.

Two aspects to making a GUI:
1. Placing components (buttons, text, etc.) in it.
2. Listening/responding to events

Next Lecture
Lecture notes page of course website, rows for GUI lectures: will contain guiDemo.zip. Filled with short demos of GUI features including demos for today and next lecture. Download it and look at demos in DrJava or Eclipse.

2110: GUIs: Graphical User Interfaces
Their mouse had a mean time between failure of ... a week ... it would jam up irreparably, or ... jam up on the table-- ... It had a flimsy cord whose wires would break. Steve Jobs: "... Xerox says it can’t be built for < $400, I want a $10 mouse that will never fail and can be mass produced, because it's going to be the primary interface of the computer ..." ...
Dean Hovey ... came back, "I've got some good and some bad news. Good news: we've got a new project with Apple. Bad news: I told Steve we'd design a mouse for 10 bucks."

... year later ... we ... filed ... and were granted a patent, on the electro-mechanical-optical mouse of today; ... we ended up ... [making] the mouse as invisible to people as it is today.

Steve Sachs interview on first computer with GUI: Apple Lisa (~$10K in 1982).
web.stanford.edu/dept/SUL/sites/mac/mouse0.html

GUI (Graphical User Interface)
There are three GUI packages in Java:
• AWT (Abstract or Awful Window Toolkit) — first one. Some parts are implemented not in Java but in code that depends on the platform. Came with first Java.
• Swing — a newer one, which builds on AWT as much as possible. It is "lightweight": all code written as Java classes/interfaces. Released in 97-98.
• JavaFX — completely new! Much more functionality, flexibility, but far too complicated to teach in CS2110. (Released first in 2008)

We use Swing (and parts of AWT)

Class JFrame
JFrame object: associated with a window on your monitor.
Generally, a GUI is a JFrame object with various components placed in it

Some methods in a JFrame object
hide() show() setVisible(boolean)
getX() getY() (coordinates of top-left point)
getWidth() getHeight() setLocation(int, int)
getTitle() setTitle(String)
getLocation() setLocation(int, int)

Over 100 methods in a JFrame object!

Class JFrame is in package javax.swing
Placing components in a JFrame

Layout manager: Instance controls placement of components.

JFrame layout manager default: BorderLayout.

BorderLayout layout manager: Can place 5 components:

```java
import java.awt.
import javax.swing.

/** Demonstrate placement of components in a JFrame.
Places five components in 5 possible areas:
1) a JButton in the east, 2) a JLabel in the west,
3) a JLabel in the south, 4) a JTextField in the north
5) a JTextArea in the center. */

class ComponentExample extends JFrame {
    ComponentExample(String t) {
        super(t); cp.add(new JButton("click me"), BorderLayout.EAST);
        add(new JTextField("type here", 22), BorderLayout.NORTH);
        add(new JCheckBox("I got up today"), BorderLayout.SOUTH);
        add(new JLabel("label 2"), BorderLayout.WEST);
        add(new JTextArea("type nhere", 4, 10), BorderLayout.CENTER);
        pack();
    }
}
```

ComponentExample.java

Put scrollbars around JTextArea:
ComponentExample2.java

Also try it without pack()
```java
import java.awt.;
import javax.swing.;

/** Instance has labels in east/west, JPanel with four buttons in center. */
public class PanelDemo extends JFrame {
    JPanel p = new JPanel();

    public PanelDemo() {
        super("Panel demo");
        p.add(new JButton("0"));   p.add(new JButton("1"));
        p.add(new JButton("2"));   p.add(new JButton("3"));
        add(new JLabel("east"), BorderLayout.EAST);
        add(new JLabel("west"), BorderLayout.WEST);
        add(new JLabel("    "), BorderLayout.SOUTH);
        add(p, BorderLayout.CENTER);
    }
}
```

**Panel as a container**

FlowLayout layout manager: Place any number of components. They appear in the order added, taking as many rows as necessary.

```java
import javax.swing.;
import java.awt.;

/** Demo class Box. Comment on constructor says how frame is laid out. */
public class BoxDemo extends JFrame {
    Box b = new Box(BoxLayout.X_AXIS);
    b.add(new JButton("0"));     b.add(new JButton("1"));
    b.add(new JButton("2"));     b.add(new JButton("3"));
    add(new JLabel("east"), BorderLayout.EAST);
    add(new JLabel("west"), BorderLayout.WEST);
    add(new JLabel("    "), BorderLayout.SOUTH);
    add(b, BorderLayout.CENTER);
}
```

**Class Box: a container**

BoxLayout layout manager: Place any number of components. They appear in the order added, taking only one row.

```java
public class BoxDemo2 extends JFrame {
    public BoxDemo2(String t, int n) {
        super(t);
        // Create Box b1 with n buttons.
        Box b1 = new Box(BoxLayout.Y_AXIS);
        for (int i = 0; i < n; i++)
            b1.add(new JButton("1 + i");
        // Create Box b2 with n+1 buttons.
        Box b2 = ...;
        // Create Box b3 with n+2 buttons.
        Box b3 = ...
        // Create horizontal box b containing b1, b2, b3
        Box b = new Box(BoxLayout.X_AXIS);
        b.add(b1);
        b.add(b2);
        b.add(b3);
        add(b, BorderLayout.CENTER);
    }
}
```

**Simulate BoxLayout Manager in a JFrame**

To simulate using a BoxLayout manager for a JFrame, create a Box and place it as the sole component of the JFrame:

```java
JFrame jf = new JFrame("title");
Box b = new Box(BoxLayout.X_AXIS);
Add components to b;
if.add(b, BorderLayout.CENTER);
```

1. Start developing a GUI by changing an already existing one. A lot of details. Hard to get all details right when one starts from scratch and has little idea about the Java GUI package.

1. Showed how to place components in a GUI. Next time: how to "listen" to things like button clicks in a GUI.