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."


---

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

A 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:

- `setVisible(boolean)`
- `show()`
- `getX()`  `getY()`  `setLocation(int, int)`
- `getTitle()`  `setTitle(String)`
- `getLocation()`  `setLocation(int, int)`

Over 100 methods in a JFrame object!

Class JFrame is in package `javax.swing`

---

Putting components in a JFrame

```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. */

public class ComponentExample extends JFrame {
    /** Constructor: a window with title t and 5 components */
    public ComponentExample(String t) {
        super(t); cp.add(new JButton("Click me"), BorderLayout.EAST);
        cp.add(new JLabel("label 2"), BorderLayout.SOUTH);
        cp.add(new JLabel("center"), BorderLayout.CENTER);
        cp.add(new JLabel("north"), BorderLayout.NORTH);
        cp.setVisible(true);
    }

    public static void main(String[] args) {
        new ComponentExample("Custom Example").setVisible(true);
    }
}
```

---

JFrameDemo.java

```java
import java.awt.*; import javax.swing.*;
/** Demonstrate placement of components in a JFrame. */

public class JFrameDemo {
    public static void main(String[] args) {
        ComponentExample ce = new ComponentExample("Custom Example");
        ce.setVisible(true);
    }
}
```

---

2110: GUIs: Graphical User Interfaces

- **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. TODAY
2. Listening/responding to events Next Lecture

---

Placing components in a JFrame

Layout manager: Instance controls placement of components.

BorderLayout layout manager: Can place 5 components:

```java
public class ComponentExample2 extends ComponentExample {
    public ComponentExample2(String t) {
        super(t); cp.add(new JButton("..."), BorderLayout.EAST);
        cp.add(new JLabel("label 2"), BorderLayout.SOUTH);
        cp.add(new JLabel("center"), BorderLayout.CENTER);
        cp.add(new JLabel("north"), BorderLayout.NORTH);
        cp.pack();
    }
}
```

---

Class JFrame

```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. */

public class ComponentExample extends JFrame {
    /** Constructor: a window with title t and 5 components */
    public ComponentExample(String t) {
        super(t); cp.add(new JButton("click me"), BorderLayout.EAST);
        cp.add(new JTextField("type here", 22), BorderLayout.NORTH);
        cp.add(new JCheckBox("I got up today"), BorderLayout.SOUTH);
        cp.add(new JTextArea("type here", 4, 10), BorderLayout.CENTER);
        cp.pack();
    }

    public static void main(String[] args) {
        ComponentExample ce = new ComponentExample("Custom Example");
        ce.pack();
    }
}
```

---

GUI (Graphical User Interface)

We use Swing (and parts of AWT)

---

Class JFrame

```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. */

public class ComponentExample extends JFrame {
    /** Constructor: a window with title t and 5 components */
    public ComponentExample(String t) {
        super(t); cp.add(new JButton("click me"), BorderLayout.EAST);
        cp.add(new JTextField("type here", 22), BorderLayout.NORTH);
        cp.add(new JCheckBox("I got up today"), BorderLayout.SOUTH);
        cp.add(new JLabel("label 2"), BorderLayout.WEST);
        cp.add(new JTextArea("type here", 4, 10), BorderLayout.CENTER);
        cp.pack();
    }

    public static void main(String[] args) {
        ComponentExample ce = new ComponentExample("Custom Example");
        ce.pack();
    }
}
```

---

Class JFrame

```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. */

public class ComponentExample extends JFrame {
    /** Constructor: a window with title t and 5 components */
    public ComponentExample(String t) {
        super(t); cp.add(new JButton("click me"), BorderLayout.EAST);
        cp.add(new JTextField("type here", 22), BorderLayout.NORTH);
        cp.add(new JCheckBox("I got up today"), BorderLayout.SOUTH);
        cp.add(new JLabel("label 2"), BorderLayout.WEST);
        cp.add(new JTextArea("type here", 4, 10), BorderLayout.CENTER);
        cp.pack();
    }

    public static void main(String[] args) {
        ComponentExample ce = new ComponentExample("Custom Example");
        ce.pack();
    }
}
```
public class BoxDemo extends JFrame {
    public PanelDemo() {
        super("Box demo");
        Box b = new Box(BoxLayout.X_AXIS);
        b.add(new JButton("0"));  b.add(new JButton("1"));
        add(new JLabel("a"), BorderLayout.EAST);
        add(new JLabel("b"), BorderLayout.WEST);
        add(new JLabel(""), BorderLayout.SOUTH);
        add(b, BorderLayout.CENTER);
        pack();
    }
}

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

Basic Components

Component
Button, Canvas
Checkbox, Choice
Label, List, Scrollbar
TextArea, TextField
Container
JComponent
AbstractButton
JButton
JToolBarButton
JToggleButton
RadioBitton
JLabel
JList
JOptionPane, JPanel
JTextField, JTextArea
JPopupMenu, JScrollBar, JSlider

Component: Something that can be placed in a GUI window. These are the basic ones used in GUIs.

Note the use of subclasses to provide structure and efficiency. For example, there are two kinds of JToggleButton, so that class has two subclasses.

Component: Something that can be placed in a GUI window. These are instances of certain classes, e.g.
JButton, JButton - Clickable button
JLabel, Label - Line of text
JTextArea, TextField - Field into which the user can type
JPanel, Panel - Used for graphics to contain other components
JCheckBox: Checkable box with a title
JComboBox: Menu of items, one of which can be checked
JRadioButton: Same functionality as JCheckBox
JScrollPane: Scrolls bars around a JText Area
Container: Can contain other components
Box: Can contain other components

Packages -- Components

javax.swing: in Swing, with text fields, etc. Components are more flexible.
javax.swing has a better way of listening to buttons, text fields, etc. Components are more flexible.

Packages that contain classes that deal with GUIs:

java.awt: Old package. javax.swing: New package.

javax.swing has a better way of listening to buttons, text fields, etc. Components are more flexible.

1. Look at AreaExample to see how to get scroll bars.
2. Look at BorderDemo to demo radio buttons, ButtonGroup, and borders.
3. Look at CheckBoxExample.
4. Look at ColorChooserExample.
5. Look at ComboBoxExample.
6. Look at SliderExample
7. Look at TemperatureSlider.
public class BoxDemo2 extends JFrame {  
/** Constructor: frame with title t and 3 columns with n, n+1, and n+2 buttons. */  
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= i+1)  
      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);  
   pack();  show();  
}  

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:  

JFramejf = new JFrame("title");  
Box b = new Box(BoxLayout.X_AXIS);  
Add components to b;  
jf.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.