Constructor for View Class
Tasks To Be Performed

• A simple constructor for the view class has four main jobs:
  1. Create the JFrame being extended
  2. Set up the GUI widgets to be used and “lay out” these widgets in the main application window
  3. Set up the observers by registering (in our examples) the object being constructed, i.e., this, with each of the GUI widgets that might have events of interest to the application
  4. Start the main application window
1: Create the JFrame

super("Simple GUI Demo");
External (GUI) Effect

this
External (GUI) Effect

Nothing illustrated in these slides actually becomes visible to the user until later!
this.inputText = new JTextArea("", LINES_IN_TEXT_areas, LINE_LENGTHS_IN_TEXT_areas);
...
JScrollPane inputTextScrollPane = new JScrollPane(this.inputText);
2: Set Up GUI Widgets (Text)...

```java
this.inputText = new JTextArea("", LINES_IN_TEXT AREAS, LINE_LENGTHS_IN_TEXT AREAS);
...
JScrollPane inputTextScrollPane =
    new JScrollPane(this.inputText);
```
External (GUI) Effect

```
this.inputText
```

```
inputTextScrollPane
```
The scrollbars in `JScrollPane`s actually arise only when needed; not needed yet, but illustrated here.
2: Set Up GUI Widgets (Buttons)...

```java
this.copyButton =
    new JButton("Copy Input");
```
External (GUI) Effect

\texttt{this.copyButton}

Copy Input
2: ... and Lay Out GUI Widgets

```java
JPanel buttonPanel =
    new JPanel(
        new GridLayout(
            ROWS_IN_BUTTON_PANEL_GRID,
            COLUMNS_IN_BUTTON_PANEL_GRID));

buttonPanel.add(this.resetButton);
buttonPanel.add(this.copyButton);
```
2: ... and Lay Out GUI Widgets

```java
JPanel buttonPanel =
    new JPanel(
        new GridLayout(
            ROWS_IN_BUTTON_PANEL_GRID,
            COLUMNS_IN_BUTTON_PANEL_GRID));
...
buttonPanel.add(this.resetButton);
buttonPanel.add(this.copyButton);
```
External (GUI) Effect
2: ... and Lay Out GUI Widgets

```java
this.setLayout(
    new GridLayout(
        ROWS_IN_THIS_GRID,
        COLUMNS_IN_THIS_GRID));
...
this.add(inputTextScrollPane);
this.add(buttonPanel);
this.add(outputTextScrollPane);
```
2: ... and Lay Out GUI Widgets

```java
this.setLayout(
    new GridLayout(
        ROWS_IN_THIS_GRID,
        COLUMNS_IN_THIS_GRID));
...
this.add(inputTextScrollPane);
this.add(buttonPanel);
this.add(outputTextScrollPane);
```
2: ... and Lay Out GUI Widgets

```java
this.setLayout(new GridLayout(ROWS_IN_THIS_GRID, COLUMNS_IN_THIS_GRID));
...
this.add(inputTextScrollPane);
this.add(buttonPanel);
this.add(outputTextScrollPane);
```

Remember:
the two buttons
are in this panel.
External (GUI) Effect
External (GUI) Effect

Remember: none of this is visible to the user yet.
3: Set Up the Observers

```java
this.resetButton.addActionListener(this);
this.copyButton.addActionListener(this);
```
Internal (non-GUI) Effect

- resetButton
- copyButton (and other widgets)
- this (DemoView)
4: Start the Main Window

```java
this.pack();
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
```
External (GUI) Effect: Now Visible

This
The only code you wrote that executes now is the callback method for the two buttons: `this.actionPerformed`. 