

# Constructor for View Class



# Tasks To Be Performed

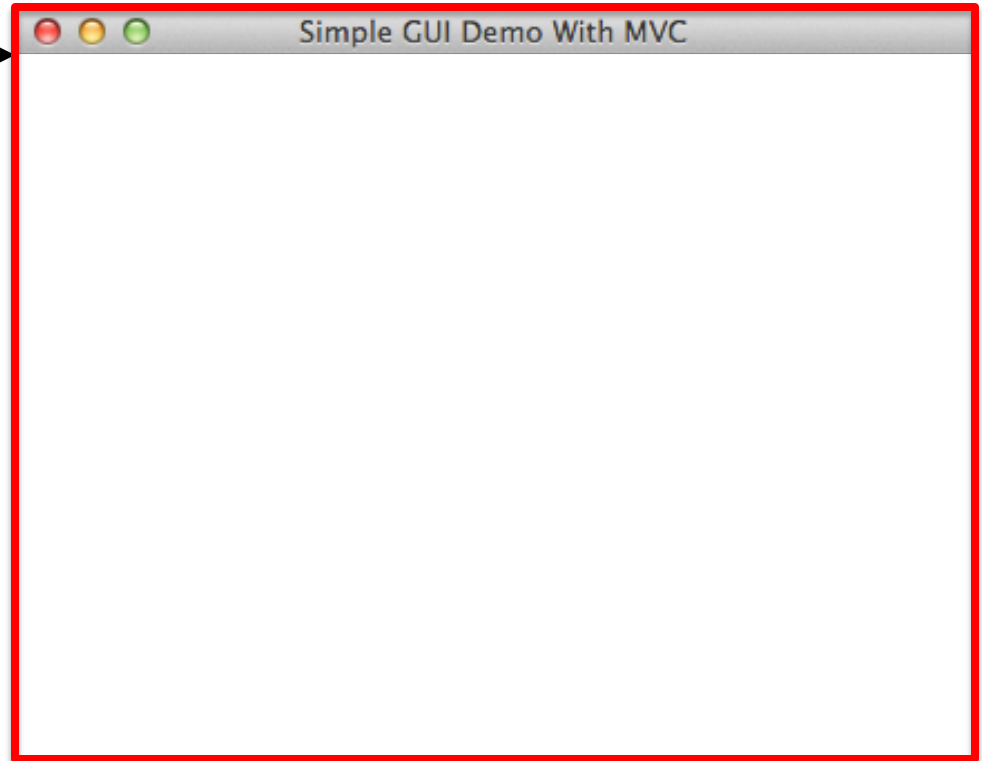
- A simple constructor for the **view** class has four main jobs:
  1. Create the `JFrame` being extended
  2. Set up the GUI widgets to be used and “lay out” these widgets in the main application window
  3. Set up the observers by registering (in our examples) the object being constructed, i.e., **this**, 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");
```

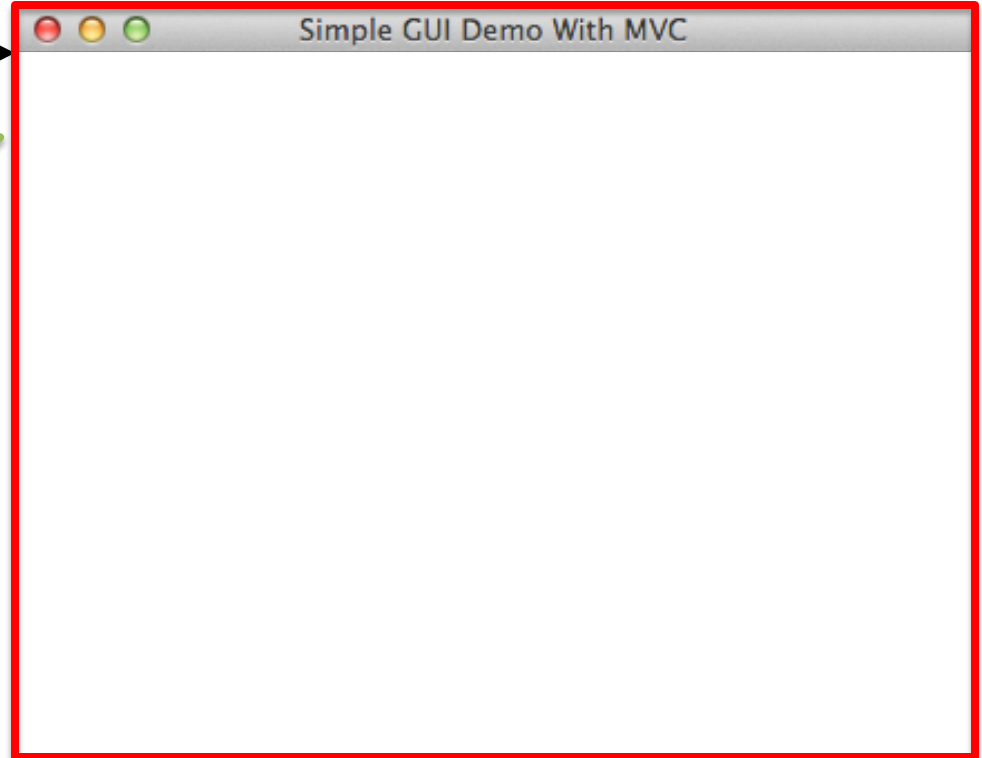
# External (GUI) Effect

**this**



# External (GUI) Effect

**this**



Nothing illustrated in these slides actually becomes visible to the user until later!

## 2: Set Up GUI Widgets (Text)...

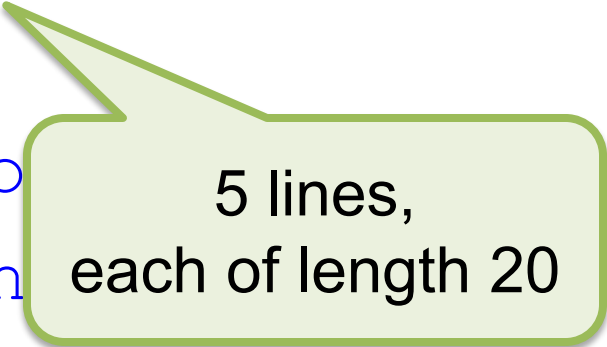
```
this.inputText = new JTextArea("",  
    LINES_IN_TEXT_AREAS,  
    LINE_LENGTHS_IN_TEXT_AREAS);  
...  
JScrollPane inputTextScrollPane =  
    new JScrollPane(this.inputText);
```

## 2: Set Up GUI Widgets (Text)...

```
this.inputText = new JTextArea("",  
    LINES_IN_TEXT_AREAS,  
    LINE_LENGTHS_IN_TEXT_AREAS);
```

...

```
JScrollPane inputTextScro  
new JScrollPane(this.in
```



5 lines,  
each of length 20

# External (GUI) Effect

`this.inputText`



`inputTextScrollPane`



# 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)...

```
this.copyButton =  
    new JButton("Copy Input");
```

# External (GUI) Effect

`this.copyButton`



## 2: ... and Lay Out GUI Widgets

```
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

```
JPanel buttonPanel =  
    new JPanel(  
        new GridLayout(  
            ROWS_IN_BUTTON_PANEL_GRID,  
            COLUMNS_IN_BUTTON_PANEL_GRID) );  
...  
buttonPanel.add( this.resetButton );  
buttonPanel.add( this.copyButton );
```



1 row,  
2 columns

# External (GUI) Effect

buttonPanel

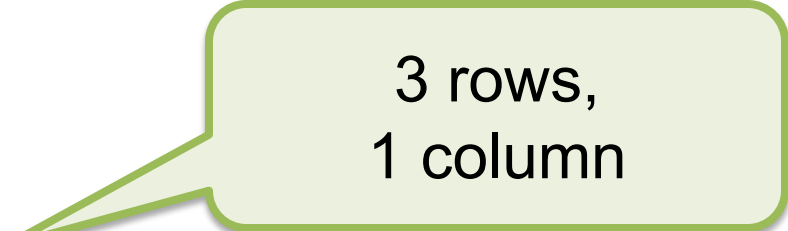


## 2: ... and Lay Out GUI Widgets

```
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

```
this.setLayout(  
    new GridLayout(  
        ROWS_IN_THIS_GRID,  
        COLUMNS_IN_THIS_GRID));
```



3 rows,  
1 column

...

```
this.add(inputTextScrollPane);  
this.add(buttonPanel);  
this.add(outputTextScrollPane);
```



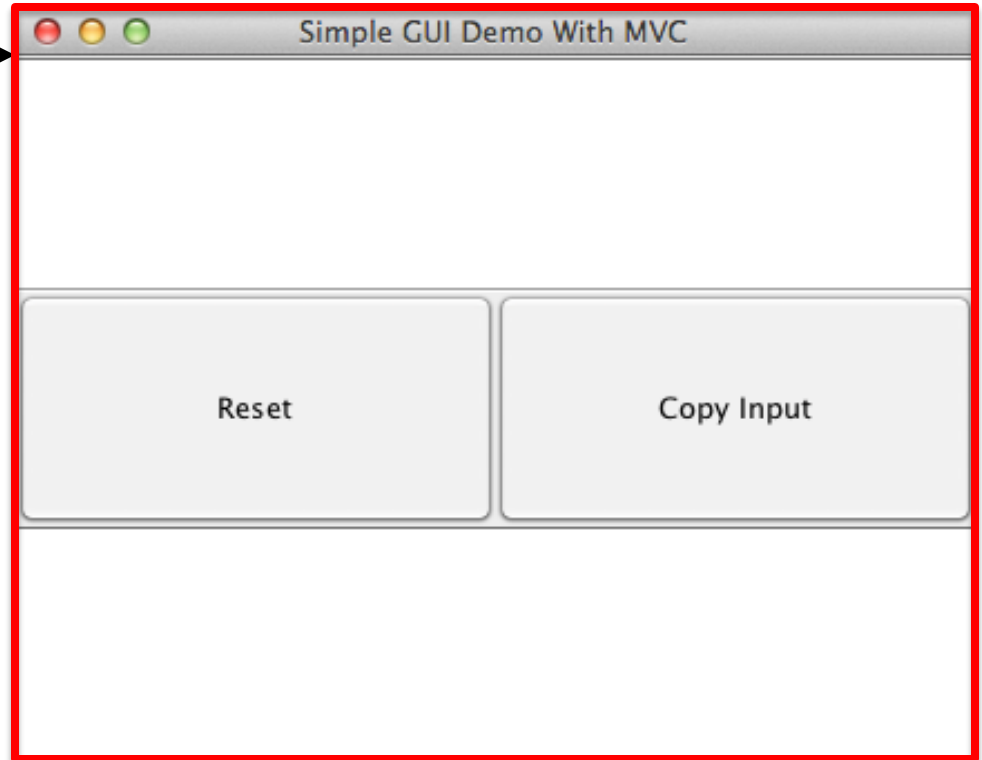
## 2: ... and Lay Out GUI Widgets

```
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.

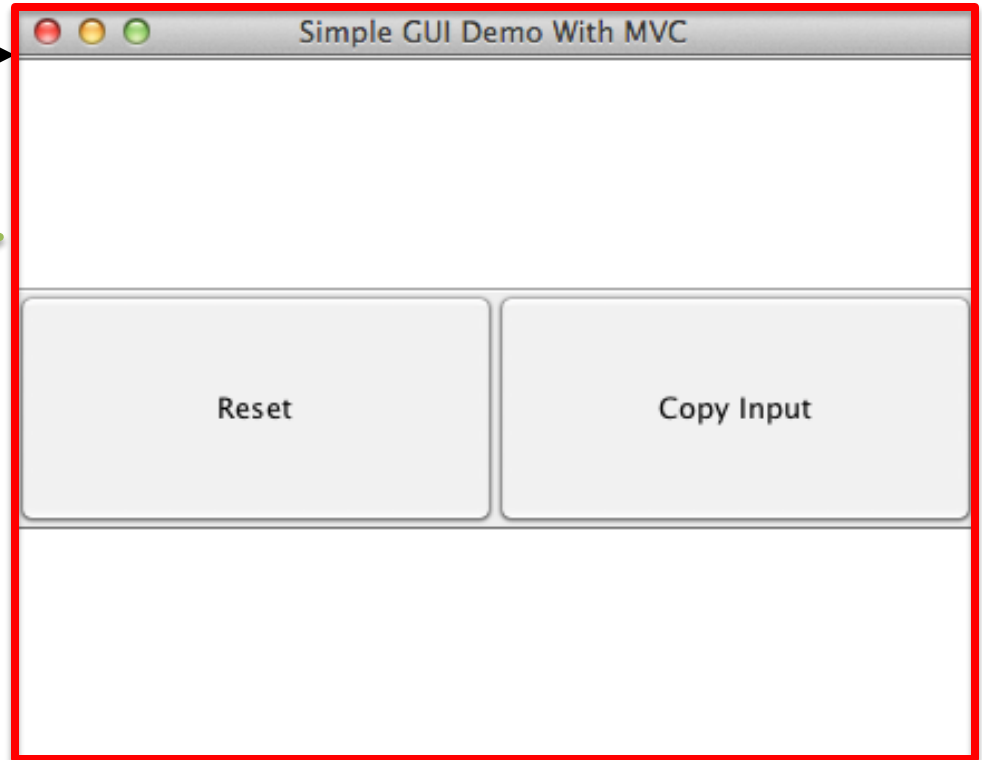
# External (GUI) Effect

**this**



# External (GUI) Effect

this

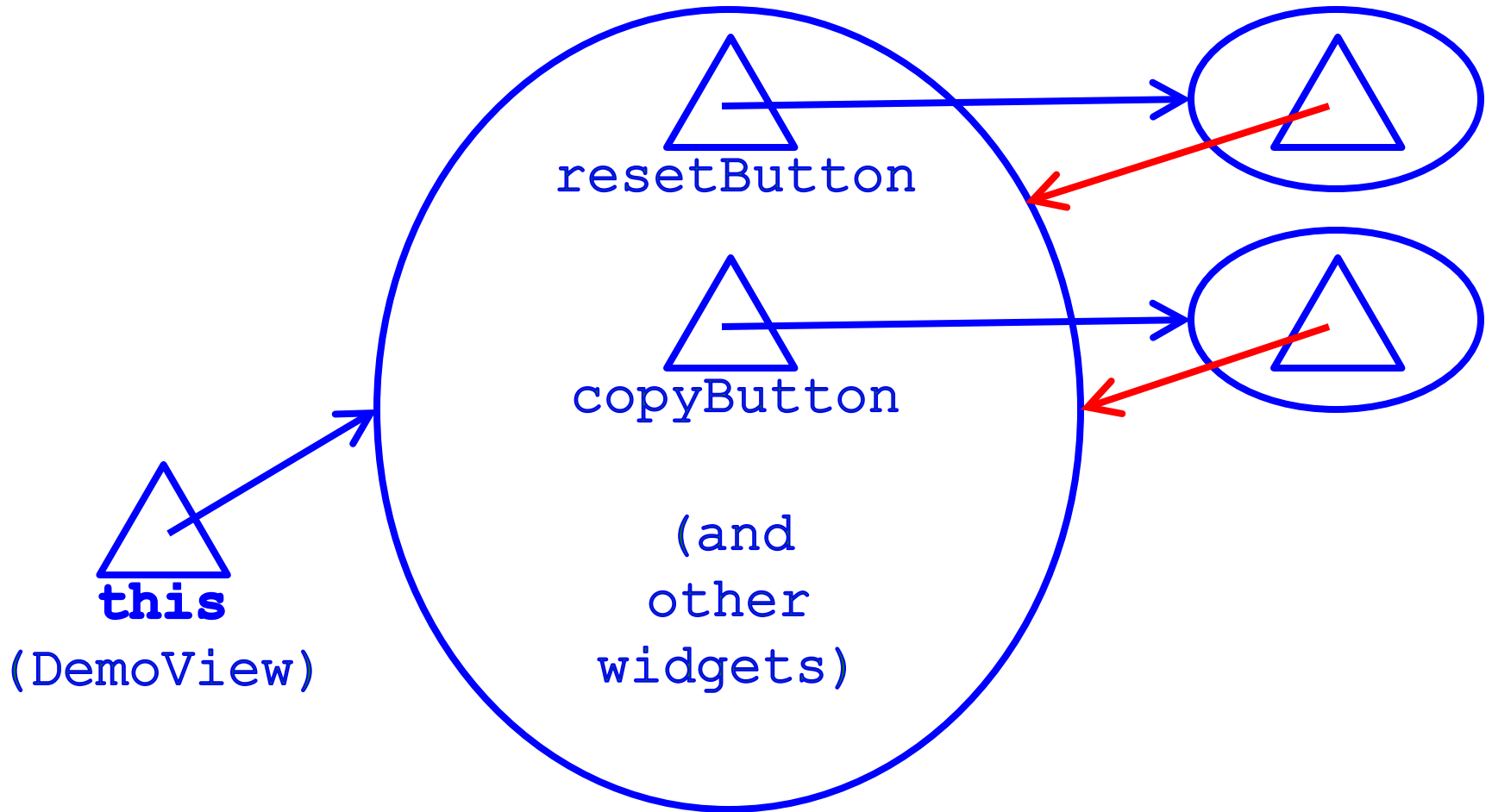


Remember:  
none of this is  
visible to the  
user yet.

# 3: Set Up the Observers

```
this.resetButton.addActionListener(this);  
this.copyButton.addActionListener(this);
```

# Internal (non-GUI) Effect

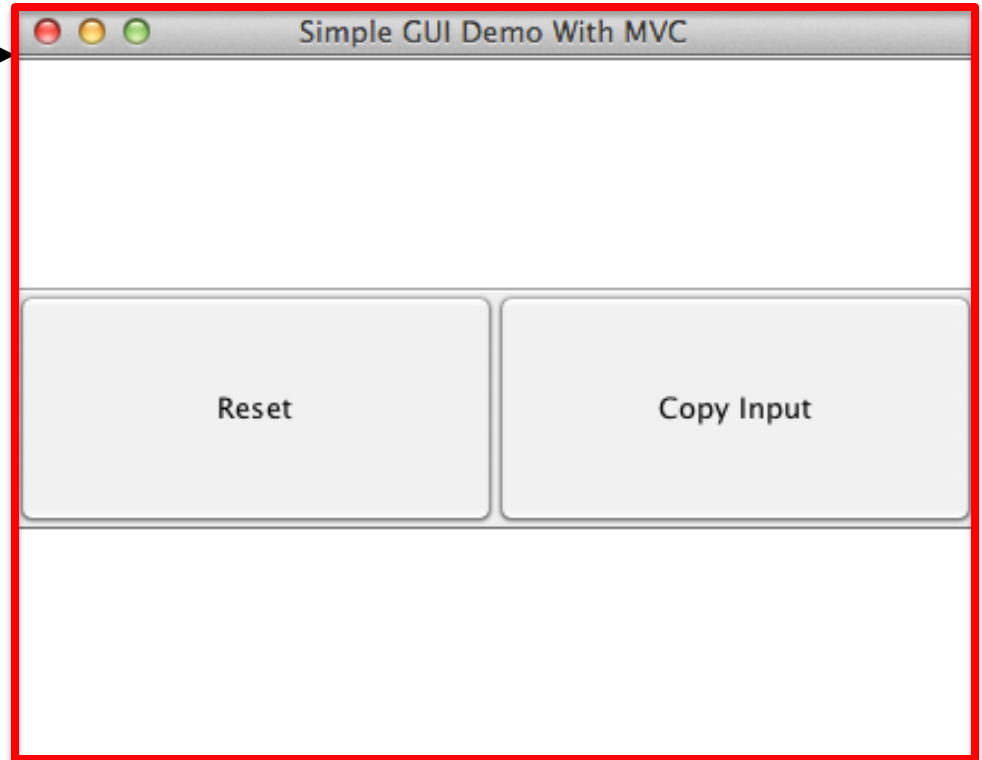


# 4: Start the Main Window

```
this.pack();  
this.setDefaultCloseOperation(  
    JFrame.EXIT_ON_CLOSE);  
this.setVisible(true);
```

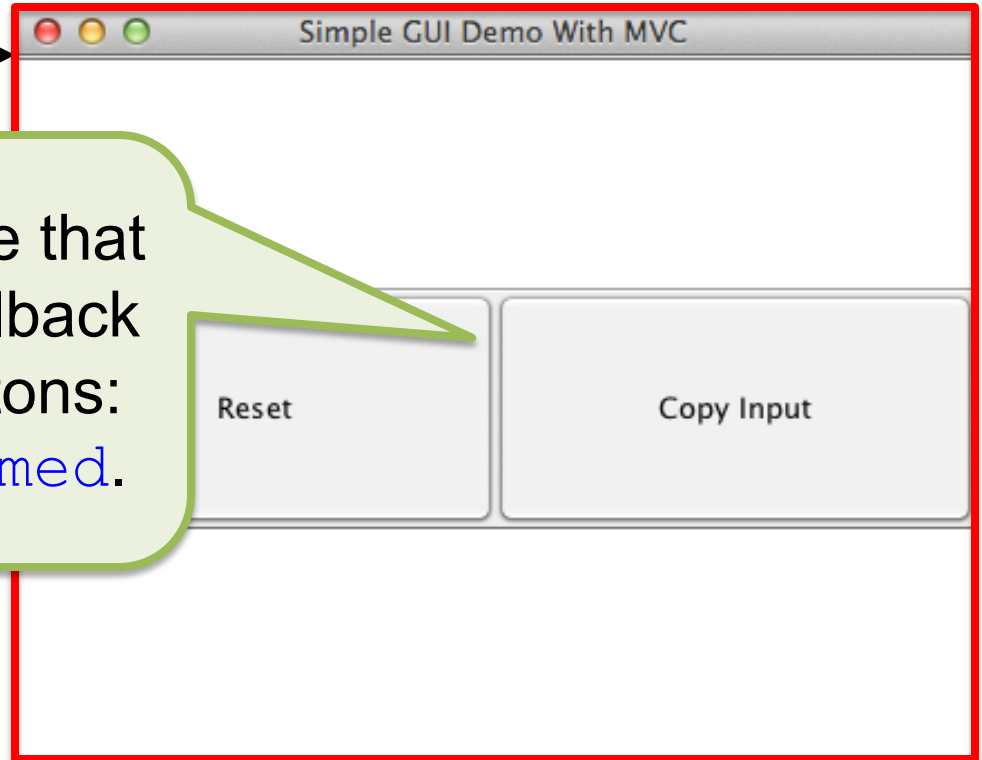
# External (GUI) Effect: Now Visible

**this**



# External (GUI) Effect: Now Visible

`this`



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