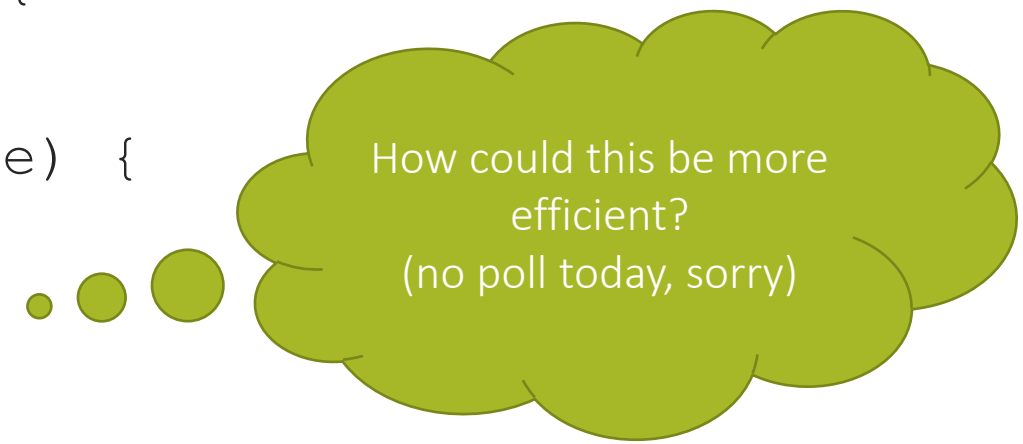
Typical submitted code (only slightly exaggerated)

```
public boolean isListEmpty() {
    boolean result;
    if (list.isEmpty() == true) {
        result = true;
    } else {
        result = false;
    }
    return result;
}
```



Modification 1 – just return!

```
public boolean isListEmpty() {
    boolean result;
    if (list.isEmpty() == true) {
        result = return true;
    } else {
        result = return false;
    }
    return result;
}
```

Modification 2 – remove “== true”

```
public boolean isListEmpty() {
    boolean result;
    if (list.isEmpty() == true) {
        result = return true;
    } else {
        result = return false;
    }
    return result;
}
```

Modification 3 – remove if/else

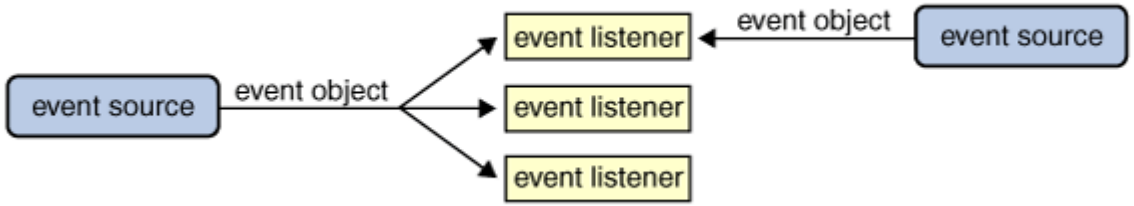
```
public boolean isListEmpty() {
    return list.isEmpty();
}
```

Today's content: Swing

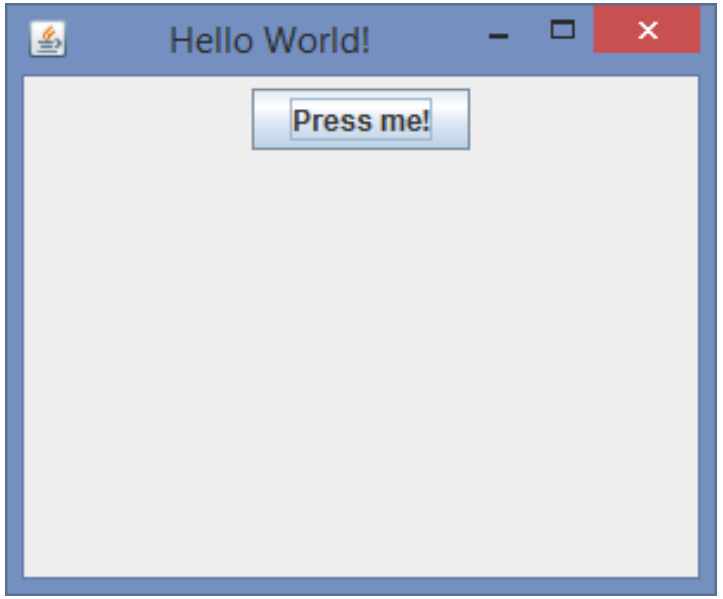
Laying out a GUI – see video/slides for some details

Writing event handlers

- Every GUI widget sends an **event** every time that the user interacts with it
- If you want to handle that event, you need to:
  - *Implement an appropriate **listener** interface*
  - *Add that listener to the component that sends events*
- Then, every time the use clicks on your button (or whatever), your event-handling code will be called!



Concrete example: clicking on a button

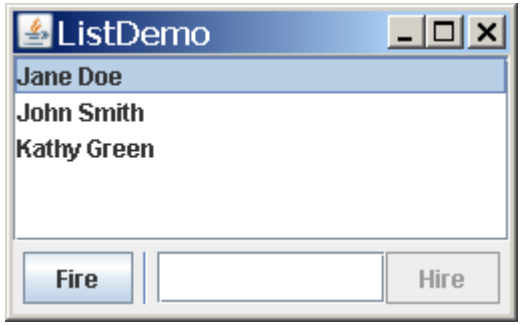



```
// Creating the button
JButton button = new JButton (“Press me!”);

// Implementing the listener
public class A implements ActionListener {
    public void actionPerformed (ActionEvent e) {
        System.out.println (“Hello”);
    }
}

// Connecting the two together
button.addActionListener (new A());
```

Modified Model-View-Controller in practice



https://docs.oracle.com/javase/tutorial/uiswing/components/list.html