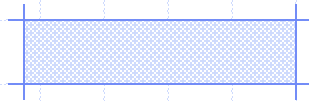


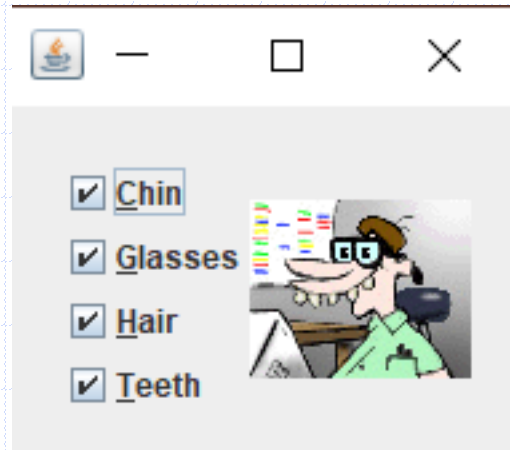
Event Handling

Placing Checkboxes



Placing a Checkbox

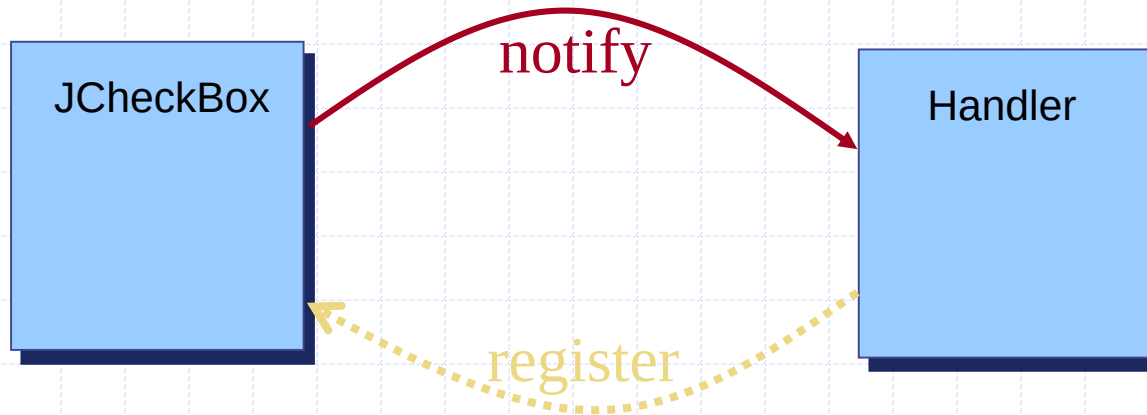
- A **JCheckBox** object is a GUI component that represents a CheckBox.
- Here's an example of how we place a Checkboxes.



Connecting Source and Listener

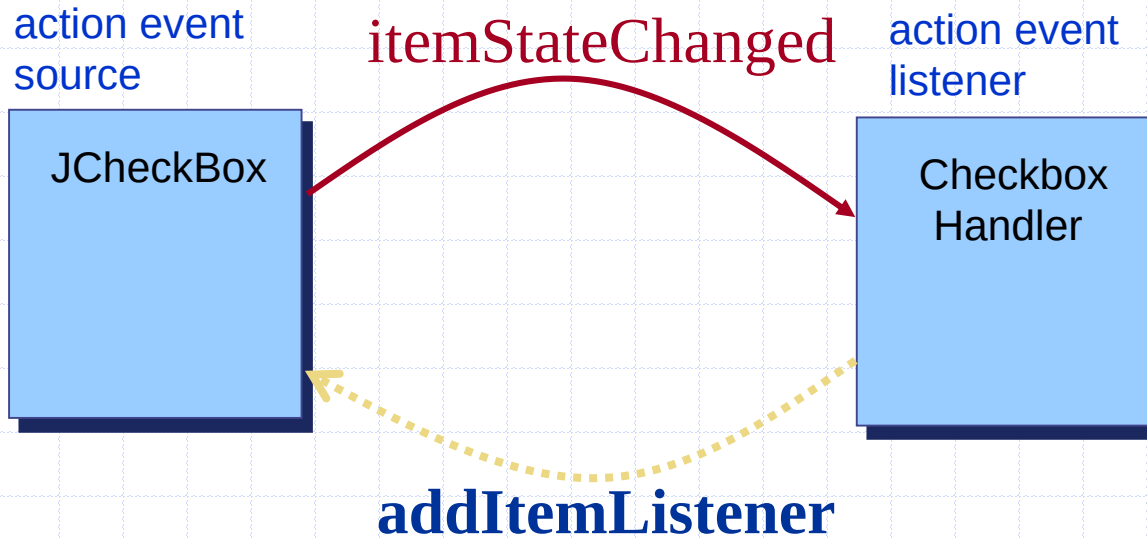
event source

event listener



A listener must be **registered** to a event source. Once registered, it will get **notified** when the event source generates events.

Handling Action Events



```
JCheckBox chinButton = new JCheckBox("Chin");  
  
chinButton.addItemListener(eventHandler);
```

ItemListener Interface

- When we call the **addItemListener** method of an event source, we must pass an instance of a class that implements the **ItemListener** interface.
- The ItemListener interface includes one method named **itemStateChanged**.
- A class that implements the ItemListener interface must therefore provide the method body of **itemStateChanged**.
- Since itemStateChanged is the method that will be called when an action event is generated, this is the place where we put a code we want to be executed in response to the generated events.

The CheckBox Handler

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

class CheckBoxnHandler implements ItemListener {
    . . .
    public void itemStateChanged(ItemEvent e) {
        Object checkedButton = e.getItemSelectable();

        // do something
    }
}
```