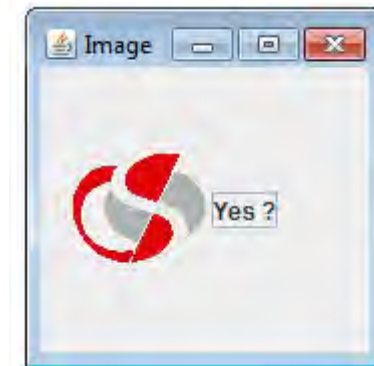


## Selection Components – Images with JCheckBox



If an icon is specified with JCheckBox it is displayed but the check box area itself is not longer available, this is the case whether the Developer includes any text or not ...

```
ImageIcon seer = new ImageIcon("c:/course/seer.gif");  
JCheckBox check = new JCheckBox("Yes ?", seer);
```



The Developer should use JCheckBoxMenuItem ...

```
import javax.swing.JCheckBoxMenuItem;  
// Need to import the above
```

```
JCheckBoxMenuItem check1 =  
    new JCheckBoxMenuItem("Yes ?", seer);
```



## Selection Components – ItemListener with JCheckBox

---



If the Developer needs to test what the state of the JCheckBox has now been changed to, then the *getStateChange()* method is employed, this is obviously more sophisticated than merely detecting that a change has taken place ...

```
ItemListener checkone = new ItemListener() {  
    public void itemStateChanged(ItemEvent event)  
    {  
        if (event.getStateChange() == ItemEvent.SELECTED)  
            System.out.println("Selected");  
        if (event.getStateChange() == ItemEvent.DESELECTED)  
            System.out.println("DeSelected");  
    }  
};
```

## Selection Components – ItemListener with JCheckBox

---



When the ItemListener is used for more than one component, the Developer can test which object 'fired' this instance of the Listener using the *getItemSelectable()* method

```
ItemListener checkone = new ItemListener() {  
    public void itemStateChanged(ItemEvent event)  
    {  
        Object item = event.getItemSelectable();  
        if (item == check1)  
            {if (event.getStateChange() == ItemEvent.SELECTED)  
                System.out.println("Selected");
```