SWING - ITEMLISTENER INTERFACE

The class which processes the ItemEvent should implement this interface.The object of that class must be registered with a component. The object can be registered using the addItemListener method. When the action event occurs, that object's itemStateChanged method is invoked.

Interface declaration

Following is the declaration for **java.awt.event.ItemListener** interface:

```
public interface ItemListener
   extends EventListener
```

Interface methods

S.N.	Method & Description
1	**void itemStateChanged*ItemEvente*** Invoked when an item has been selected or deselected by the user.

Methods inherited

This interface inherits methods from the following interfaces:

- java.awt.EventListener

ItemListener Example

Create the following java program using any editor of your choice in say **D:/ > SWING > com > tutorialspoint > gui >**

SwingListenerDemo.java

```
package com.tutorialspoint.gui;

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

public class SwingListenerDemo {
   private JFrame mainFrame;
   private JLabel headerLabel;
   private JLabel statusLabel;
   private JPanel controlPanel;

   public SwingListenerDemo(){
      prepareGUI();
   }

   public static void main(String[] args){
      SwingListenerDemo  swingListenerDemo = new SwingListenerDemo();
      swingListenerDemo.showItemListenerDemo();
   }

   private void prepareGUI(){
      mainFrame = new JFrame("Java SWING Examples");
      mainFrame.setSize(400,400);
      mainFrame.setLayout(new GridLayout(3, 1));
```

```
      headerLabel = new JLabel("",JLabel.CENTER );
      statusLabel = new JLabel("",JLabel.CENTER);

      statusLabel.setSize(350,100);
      mainFrame.addWindowListener(new WindowAdapter() {
         public void windowClosing(WindowEvent windowEvent){
         System.exit(0);
         }
      });
      controlPanel = new JPanel();
      controlPanel.setLayout(new FlowLayout());

      mainFrame.add(headerLabel);
      mainFrame.add(controlPanel);
      mainFrame.add(statusLabel);
      mainFrame.setVisible(true);
   }

   private void showItemListenerDemo(){
      headerLabel.setText("Listener in action: ItemListener");
      JCheckBox chkApple = new JCheckBox("Apple");
      JCheckBox chkMango = new JCheckBox("Mango");
      JCheckBox chkPeer = new JCheckBox("Peer");

      chkApple.addItemListener(new CustomItemListener());
      chkMango.addItemListener(new CustomItemListener());
      chkPeer.addItemListener(new CustomItemListener());

      controlPanel.add(chkApple);
      controlPanel.add(chkMango);
      controlPanel.add(chkPeer);
      mainFrame.setVisible(true);
   }


   class CustomItemListener implements ItemListener {
      public void itemStateChanged(ItemEvent e) {
         statusLabel.setText(((JCheckBox)e.getItem()).getText()
         +" Checkbox: "
         + (e.getStateChange()==1?"checked":"unchecked"));
      }
   }
}
```

Compile the program using command prompt. Go to **D:/ > SWING** and type the following command.

```
D:\SWING>javac com\tutorialspoint\gui\SwingListenerDemo.java
```

If no error comes that means compilation is successful. Run the program using following command.

```
D:\SWING>java com.tutorialspoint.gui.SwingListenerDemo
```

Verify the following output