

* Event Handling in Java :-

Changing the state of an object is known as an event.

For example:- click on button, dragging mouse etc.

o Steps of Perform Event Handling:-

for registering the component with the listener, many classes provide the registration methods.

(i) Button

```
public void addActionListener(ActionListener a) {}
```

(ii) MenuItem

```
public void addActionListener(ActionListener a) {}
```

(iii) TextField

```
public void addActionListener(ActionListener a) {}
public void addTextListener(TextListener a) {}
```

(iv) TextArea

```
public void addTextListener(TextListener a) {}
```

(v) Checkbox

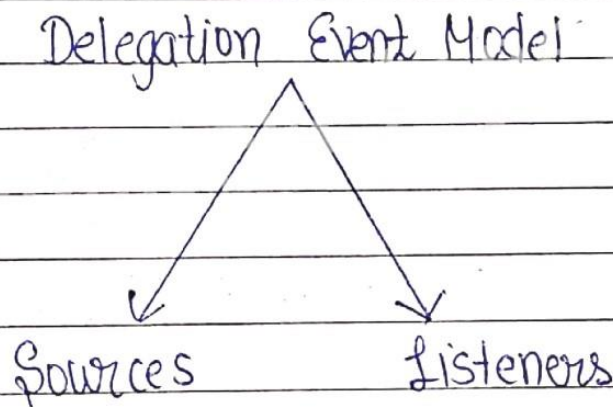
```
public void addItemListener(ItemListener a) {}
```

//_

* Delegation Event Model :-

- To handle the events, Java follows the Delegation Event model.
- The delegation event model, which defines standard and consistent mechanisms to generate and process events.

It has Sources and Listeners :-



1. Sources :-

- It is an object on which event occurs.
- Events are generated from the source.
- There are various sources like buttons, checkboxes, list, menu-item, choice etc to generate events.

2 Listeners:-

- It is also Event handler.
- It Generates response to event.
- Listeners are basically used for handling the events generated from the source.

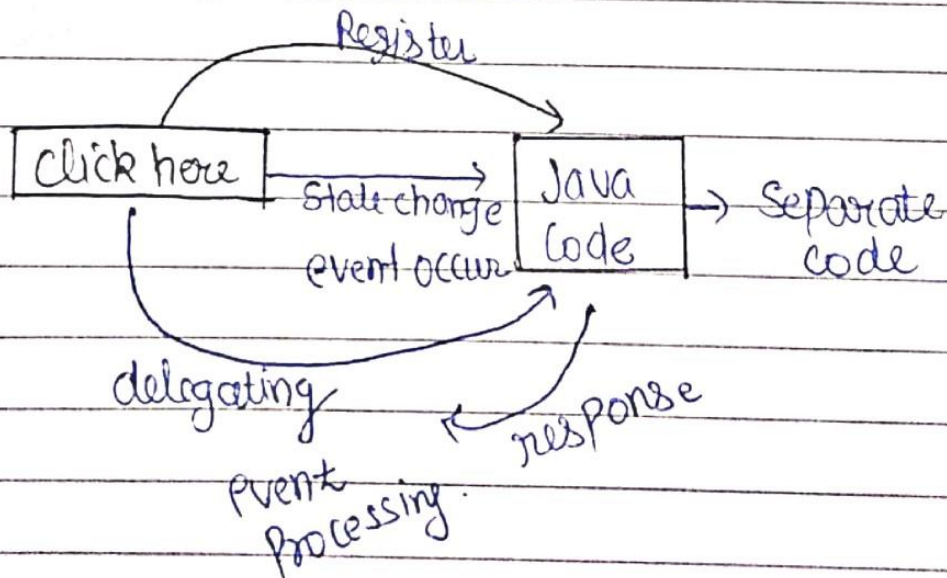
Syntax:

addTypeListener()



addKeyListener()

addActionListener()



Java Listener Interfaces

1. Action Listener :-

- It is used to handle events caused by sources like Buttons, MenuItem etc.
- The ActionListener interface is found in java.awt.event package - It has only one method

Example :-

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

```
class eventtest extends Frame implements  
ActionListener
```

```
{
```

```
    Frame f;  
    TextField tf;  
    Button b;  
    eventdemo()
```

```
{
```

```
    f = new Frame();  
    f.setLayout(new FlowLayout());  
    f.setSize(500, 500);  
    f.setVisible(true);  
    b = new Button("Click here");  
    tf = new TextField(10);
```


_ / _ / _

```
f.add(b);  
f.add(tf);  
b.addActionListener(this);
```

```
g  
public void actionPerformed(ActionEvent ae)
```

```
    {  
        String str = ae.getActionCommand();  
        tf.setText(str);
```

```
g  
public static void main (String args[])
```

```
    {  
        eventdemo e = new eventdemo();
```

```
g
```

② Java MouseListener Interface :-

- The Java MouseListener is notified whenever you change the state of mouse.
- The MouseListener interfaces is found in java.awt.event.*.

It has five Methods

```
public abstract void mouseClicked (MouseEvent e);
public abstract void mouseEntered (MouseEvent e);
public abstract void mouseExited (MouseEvent e);
public abstract void mousePressed (MouseEvent e);
public abstract void mouseReleased (MouseEvent e);
```

Example

```
import java.awt.*;
import java.awt.event.*;
public class MouseListenerExample extends Frame
implements MouseListener
{
    Label l = new Label ();

    MouseListenerExample ()
    {
        addMouseListener (this);
    }
}
```


//_

```
l. setBounds (25, 50, 100, 20);  
add (l);  
setSize (300, 300);  
setLayout (null);  
setVisible (true);
```

3

```
public void mouseClicked (MouseEvent e)  
{
```

```
l. setText ("Mouse Clicked");
```

3

```
public void mouseEntered (MouseEvent e)  
{
```

```
l. setText ("Mouse Entered");
```

3

```
public void mouseExited (MouseEvent e)  
{
```

```
l. setText ("Mouse Exit");
```

3

```
public void mouseReleased (MouseEvent e)  
{
```

```
l. setText ("Mouse Released");
```

3

```
public void mousePressed (MouseEvent e)  
{
```

```
l. setText ("Mouse Pressed");
```

3

```
public static void main (String args []) {  
    new MouseListenerExample ();
```

3, 3

③ Java KeyListener Interface :-

- The Java keyListener is notified whenever you change the state of key.
- The keyListener interface is found in java.awt.event package.
- It has three methods :-
 1. public void keyPressed (MouseEvent e) :- It is invoked when a key ^{has been} pressed.
 2. public void keyReleased (MouseEvent e) :- It is invoked when a key has been released.
 3. public void keyTyped (MouseEvent e) :- It is invoked when a key has been typed.

Example

```

import java.awt.*;
import java.awt.event.*;
public class KeyListenerExample extends Frame implements
KeyListener {
    Label l = new Label ();
    TextArea area = new TextArea ();
  
```


_ / _ / _

```
KeyListenerExample()
{
```

```
    l.setBounds(20, 50, 100, 20);
    area.setBounds(20, 50, 300, 300);
    area.addKeyListener(this);
```

```
    add(l);
    add(area);
```

```
    setSize(400, 400); // Setting the size, layout
    setLayout(null);    // and visibility of frame
    setVisible(true);
```

```
    public void keyPressed (KeyEvent e)
    {
```

```
        l.setText("Key Pressed");
    }
```

```
    public void keyReleased(KeyMouseEvent e)
    {
```

```
        l.setText("key Released");
    }
```

```
    public void keyTyped (KeyEvent e)
    {
```

```
        l.setText("key Typed");
    }
```

11

```
public static void main (String args [])
```

```
    {
```

```
        {
```

```
            {
```