**Event Handling**

Any program that uses GUI (graphical user interface) such as Java application written for windows, is event driven.

Event describes the change in state of any object.

**For Example :** Pressing a button, Entering a character in Textbox, Clicking or Dragging a mouse, etc.

**Components of Event Handling**

Event handling has three main components,

* **Events :** An event is a change in state of an object.
* **Events Source :** Event source is an object that generates an event.
* **Listeners :** A listener is an object that listens to the event. A listener gets notified when an event occurs.

**How Events are handled ?**

A source generates an Event and send it to one or more listeners registered with the source. Once event is received by the listener, they process the event and then return. Events are supported by a number of Java packages, like **java.util**, **java.awt** and **java.awt.event**.

**Important Event Classes and Interface**

|  |  |  |
| --- | --- | --- |
| **Event Classes** | **Description** | **Listener Interface** |
| **ActionEvent** | generated when button is pressed, menu-item is selected, list-item is double clicked | ActionListener |
| **MouseEvent** | generated when mouse is dragged, moved,clicked,pressed or released and also when it enters or exit a component | MouseListener |
| **KeyEvent** | generated when input is received from keyboard | KeyListener |
| **ItemEvent** | generated when check-box or list item is clicked | ItemListener |
| **TextEvent** | generated when value of textarea or textfield is changed | TextListener |
| **MouseWheelEvent** | generated when mouse wheel is moved | MouseWheelListener |
| **WindowEvent** | generated when window is activated, deactivated, deiconified, iconified, opened or closed | WindowListener |
| **ComponentEvent** | generated when component is hidden, moved, resized or set visible | ComponentEventListener |
| **ContainerEvent** | generated when component is added or removed from container | ContainerListener |
| **AdjustmentEvent** | generated when scroll bar is manipulated | AdjustmentListener |
| **FocusEvent** | generated when component gains or loses keyboard focus | FocusLis |

An action event occurs, whenever an action is performed by the user. Examples: When the user clicks a [button](https://docs.oracle.com/javase/tutorial/uiswing/components/button.html), chooses a [menu item](https://docs.oracle.com/javase/tutorial/uiswing/components/menu.html), presses Enter in a [text field](https://docs.oracle.com/javase/tutorial/uiswing/components/textfield.html). The result is that an actionPerformed message is sent to all action listeners that are registered on the relevant component.

To write an Action Listener, follow the steps given below:

1. Declare an event handler class and specify that the class either implements an ActionListener interface or extends a class that implements an ActionListener interface. For example:
2. public class MyClass implements ActionListener {
3. Register an instance of the event handler class as a listener on one or more components. For example:
4. someComponent.addActionListener(instanceOfMyClass);
5. Include code that implements the methods in listener interface. For example:
6. public void actionPerformed(ActionEvent e) {
7. ...//code that reacts to the action...

}