



SNS COLLEGE OF TECHNOLOGY
(an autonomous institution)
Coimbatore - 35



19CST102 - OBJECT ORIENTED PROGRAMMING
I YEAR / II SEMESTER

UNIT IV - DESIGNING APPLET IN JAVA
TOPIC : EVENT HANDLING

Guided by :
Mr.Selvakumar. N
AP/CSE

Presented by:
SACHIN SARTH M
(713522CS133)



Event Handling

- Event handling is fundamental to Java programming because it is used to create event driven programs eg
 - Applets
 - GUI based windows application
 - Web Application
- Event handling mechanism have been changed significantly between the original version of Java (1.0) and all subsequent versions of Java, beginning with version 1.1.
- The modern approach to handling events is based on the **delegation event model**,



Event, Event Source, Event Listener



What is an Event?

Change in the state of an object is known as event i.e. event describes the change in state of source. Events are generated as result of user interaction with the graphical user interface components. For example, clicking on a button, moving the mouse, entering a character through keyboard, selecting an item from list, scrolling the page are the activities that causes an event to happen.



What is Event Handling?

Event Handling is the mechanism that controls the event and decides what should happen if an event occurs. This mechanism have the code which is known as event handler that is executed when an event occurs. Java Uses the Delegation Event Model to handle the events. This model defines the standard mechanism to generate and handle the events. Let's have a brief introduction to this model.

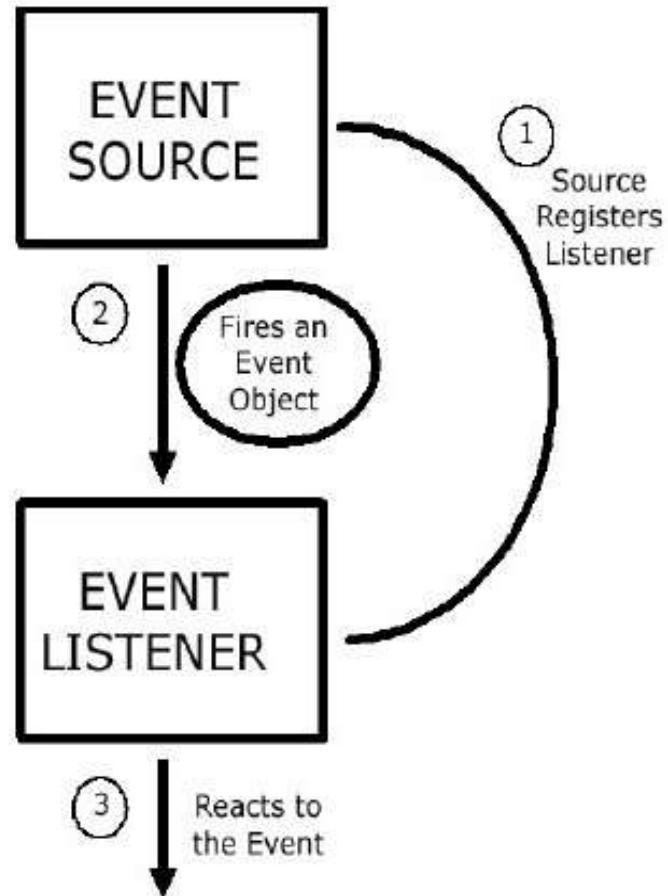
The Delegation Event Model has the following key participants namely:

Source - The source is an object on which event occurs. Source is responsible for providing information of the occurred event to it's handler. Java provides classes for source object.

Listener - It is also known as event handler. Listener is responsible for generating response to an event. From java implementation point of view the listener is also an object. Listener waits until it receives an event. Once the event is received , the listener process the event an then returns.



Delegation Event Model





Events and Event Classes

The root class is called `java.util.EventObject`. The only common feature shared by all events is a `source object`. So we find the following two methods in the `EventObject` class :

```
public Object getSource();
```

Returns the source of the event.

```
String toString( );
```

returns the string equivalent of the event.



Commonly used Event Classes in java.awt.event

Event	Class Description
ActionEvent	Generated when a button is pressed, a list item is double-clicked, or a menu item is selected.
AdjustmentEvent	Generated when a scroll bar is manipulated.
ComponentEvent	Generated when a component is hidden, moved, resized, or becomes visible.
ContainerEvent	Generated when a component is added to or removed from a container.
FocusEvent	Generated when a component gains or loses keyboard focus.
InputEvent	Abstract superclass for all component input event classes.
ItemEvent	Generated when a check box or list item is clicked; also occurs when a choice selection is made or a checkable menu item is selected or deselected.



Program to Add a Button to a Frame

```
import java.awt.*;  
import java.awt.event.*;  
public class ButtonText {  
    public static void main(String[] args) {  
        Frame frame=new Frame("Button Frame");  
        Button button = new Button("Submit");  
        frame.add(button);  
        frame.setLayout(new FlowLayout());  
        frame.setSize(200,100);  
        frame.setVisible(true);  
    }  
}
```




```
frame.addWindowListener(new WindowAdapter(){  
public void windowClosing(WindowEvent e)  
{System.exit(0);} }); }  
}
```



Three players of Delegation Event



Event source which generates the event object

Event listener which receives the event object and handles it

Event object that describes the event.