Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

SAIRAM DIGITAL RESOURCES

YEAR II SEM III

CS8392

OBJECT ORIENTED PROGRAMMING
(Common to CSE, EEE, EIE, ICE, IT)

UNIT NO 5

EVENT DRIVEN PROGRAMMING

5.3 BASICS OF EVENT HANDLING - EVENT HANDLERS

COMPUTER SCIENCE & ENGINEERING

Introduction to Event Handling

EVENTS:

- 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.
- **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.

Types of Events

- **Foreground Events :**

  Those events which require the direct interaction of user.

  They are generated as consequences of a person interacting with the graphical components in Graphical User Interface.

  Example: clicking on a button, moving the mouse, entering a character through keyboard,selecting an item from list, scrolling the page etc.

- **Background Events :**

  Those events that require the interaction of end user are known as background events.

  Operating system interrupts, hardware or software failure, timer expires, an operation completion are the example of background events.

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 Event Delegation Model to handle the events.
- This model defines the standard mechanism to generate and handle the events.

Event Handling

- **Steps involved in event handling**
  - The User clicks the button and the event is generated.
  - Now the object of concerned event class is created automatically and information about the source and the event get populated with in same object.
  - Event object is forwarded to the method of registered listener class.
  - The method is now get executed and returns.

How Event Handling works in AWT

- Implemented through Event Delegation Model
  - Model used by Java to handle user interaction with GUI components
  - Describes how your program can respond to user interaction
- Has three Components:
    - Event sources (such as buttons or scrollbars)
    - Event listeners/Handler
    - Event Object

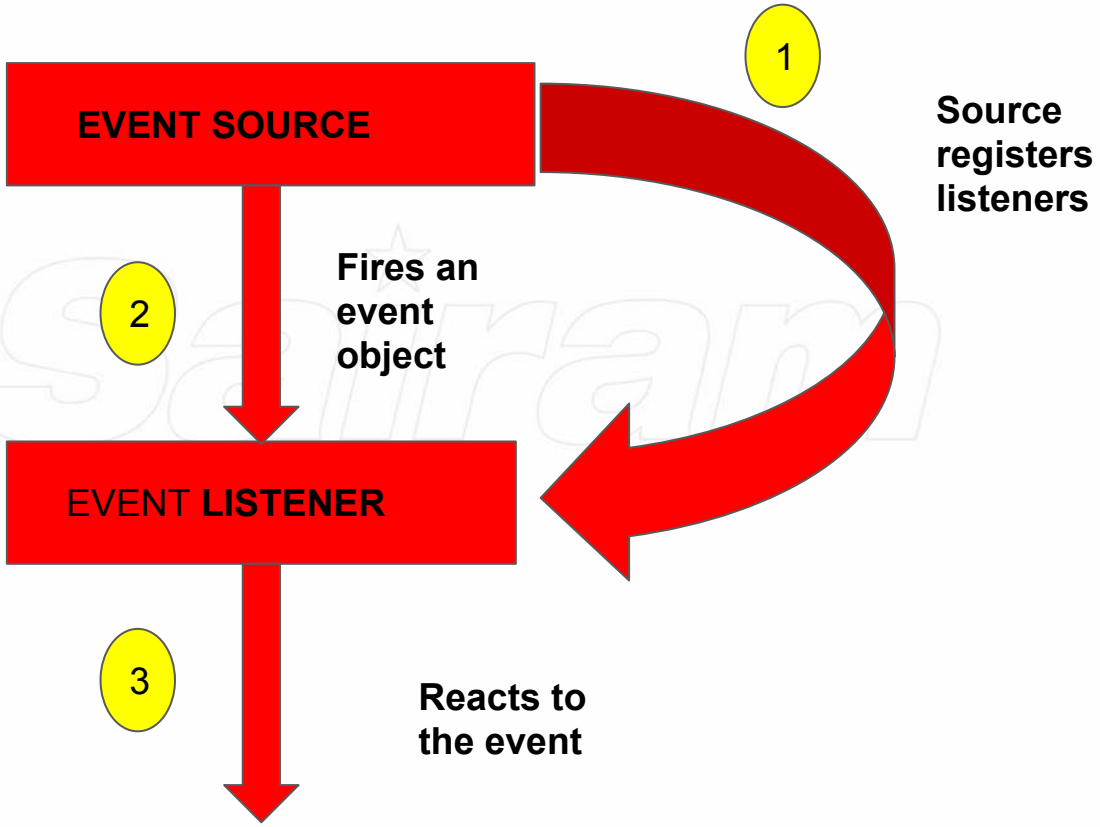
How Event Handling works in AWT

- **Event Source**
  - GUI component that generates the event
  - Example: button
- **Event Listener/Handler**
  - Receives and handles events
  - Contains business logic
  - Example: displaying information useful to the user, computing a value
- **Event Object**
  - Created when an event occurs (i.e., user interacts with a GUI component)
  - Represented by an Event class
  - Contains all necessary information about the event that has occurred
    - Type of event that has occurred
    - Source of the event

How Event Handling works in AWT

- A listener object is an instance of a class that implements a special interface called (naturally enough) a listener interface.
- An event source is an object that can register listener objects and send them event objects.
- The event source sends out event objects to all registered listeners when that event occurs.
- The listener objects will then use the information in the event object to determine their reaction to the event.

How Event Handling works in AWT



Event Classes

- Events are supported by a number of Java packages, like **java.util**, **java.awt** and **java.awt.event**.

The EventObject class

- Found in the java.util package

The AWT Event class

- An immediate subclass of EventObject
- Defined in java.awt package
- Root of all AWT-based events
- Subclasses follow this naming convention:
  - <Type> 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

Important Event Classes and Interface

Event Classes	Description	Listener Interface
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

Important Event Classes and Interface

Event Classes	Description	Listener Interface
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	FocusListener

Steps for Creating GUI Applications with Event Handling

1.Create a GUI class

Describes and displays the appearance of your GUI application

2.Create Event Listener class (a class implementing the appropriate listener interface)

Override all methods of the appropriate listener interface

Describe in each method how you would like the event to be handled

May give empty implementations for methods you don't need

3.Register the listener object with the event source

The object is an instantiation of the listener class in step 2

Use the addListener method of the event source