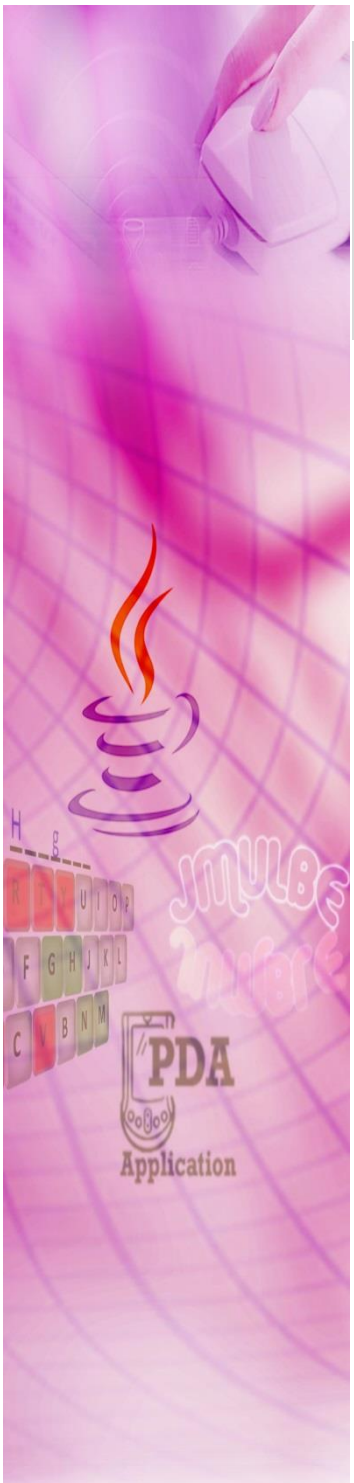

Instructor Inputs

Session 11

Session Overview

This session covers Chapter 8 of the book, Introduction in Java – SG. This session will help the students to become familiar with the event handling concepts in Java. In addition, the students will learn how to implement event handling in Java.

Exploring Java Event Model

Handling Tips

Start the discussion by asking the students how they can interact with the user interface. Elicit the responses, and then start explaining the students that the process of interaction between an application and a user. Thereafter, explain the students that the interaction is implemented in Java programs through the event model supported by the Java programming language.

Further, explain each component of the event model in detail. Thereafter, introduce the event classes and explain the various types of classes with their commonly used methods. Next, explain the event listeners and discuss the various types of event listeners with their methods. Thereafter, introduce the adapter classes and discuss why they are important. In addition, explain the event listeners supported by the Swing components with the help of the information given in the Additional Inputs topic.

Additional Inputs

The following table lists the event listeners supported by the components.

Component	***ActionListener***	***CaretListener***	***ChangeListener***	***ItemListener***	***ListSelectionListener***	***WindowListener***
`JButton`	*Yes*		*Yes*	*Yes*		
`JCheckBox`	*Yes*		*Yes*	*Yes*		
`JComboBox`	*Yes*			*Yes*		
`JDialog`						*Yes*
`JFrame`						*Yes*
`JList`					*Yes*	
`JOptionPane`						
`JRadioButton`	*Yes*		*Yes*	*Yes*		
`JTabbedPane`			*Yes*			

Component	*ActionListener*	*CaretListener*	*Change Listener*	*ItemListener*	*ListSelectionListener*	*Window Listener*
`JTextArea`		*Yes*				
`JTextField`	*Yes*	*Yes*				

The Event Listeners Supported by the Components

Implementing Events

Handling Tips

Start the discussion by explaining the process of event model with respect to a source and a listener. Thereafter, emphasize on the fact that the listener interface provides the handler methods that need to be overridden by the class that implements the listener interface. Further, explain the implementation of the action handler and action listener, mouse handler and mouse listener, and the window handler and window listener, with the help of the code given in the SG. Further, explain how to associate an event handler with the event source by using the add listener methods provided by the components that generate the respective event with the help of the given code snippet.

In addition, explain the code given in the Additional Examples topic to demonstrate the implementation of the item event by using the `ItemListener` interface and the `itemStateChanged()` method.

Additional Examples

The following code demonstrates the implementation of the item event:

```
import java.awt.FlowLayout;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

public class MultiEventDemo extends JFrame implements ItemListener
{
    JComboBox country;

    public MultiEventDemo()
    {
        setLayout(new FlowLayout());
        setSize(200,100);

        country = new JComboBox();
        country.addItem("England");
        country.addItem("China");
        country.addItem("Turkey");
        country.addItem("Russia");
```



```
        country.addItemListener(this);

        add(country);
        setVisible(true);

    }

   public void itemStateChanged(ItemEvent e)
   {
       JOptionPane.showMessageDialog(null, "You have selected " +
country.getSelectedItem());
   }

    public static void main(String[] args)
    {
        MultiEventDemo obj = new MultiEventDemo();


    }
}
```

Activity 8.1: Implementing Events

Handling Tips

Discuss the problem statement with the students.

To perform the activity, 8.1, you need to use the **GameWindow.txt** and **Menu.txt** files, which are provided at the following location in the TIRM CD:

- **Datafiles For Faculty\Activities\Chapter 08\Activity 8.1\Input Files**

The solution files, **Hangman.java**, **Menu.java**, and **GameWindow.java**, for this activity are provided at the following location in the TIRM CD:

- **Datafiles For Faculty\Activities\Chapter 08\Activity 8.1\Solution**

FAQs

- *What are the events that a component can generate?*

  Ans: A component can generate different types of events. The simplest way to know what event a component generates is to look for methods in the component's class, which has the form, `add<event_name>Listener()`. For example, if a component has an `addMouseListener()` method, the component generates `MouseEvent`.

- *Can a component be associated with more than one listener?*

  Ans: Yes, a component can be associated with more than one listener. When an event occurs, the respective listener is notified.

- *Can a component generate more than one type of event?*

  Ans: Yes, all the components can generate more than one type of event.