Unit 3

Event Handling

Event Handling:

Event:

- Change in the state of an object is known as **Event**, i.e., event describes the change in the state of the source.
- Events are generated as a 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 the list, and scrolling the page are the activities that causes an event to occur.

Event Handling :

- Event Handling is the mechanism that controls the event and decides what should happen if an event occurs.
- This mechanism has a code which is known as an 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.

The Delegation Event Model has the following key participants.

1. **Source** − The source is an object on which the event occurs. Source is responsible for providing information of the occurred event to it's handler. Java provide us with classes for the source object.

2. **Listener** − It is also known as event handler. The listener is responsible for generating a response to an event. The listener waits till it receives an event. Once the event is received, the listener processes the event and then returns.

Java Event classes and listener interfaces :

- Event classes represent the event.
- Event listeners represent the interfaces responsible to handle events.

Event Classes	Listener Interfaces
ActionEvent	ActionListener
MouseEvent	MouseListener
FocusEvent	FocusListener
KeyEvent	KeyListener
ItemEvent	ItemListener
WindowEvent	WindowListener

ActionListener :

- The Java ActionListener is notified whenever you click on the button or menu item.
- It is notified against ActionEvent.
- The ActionListener interface is found in java.awt.event package.
- It has only one method: actionPerformed().

- The signature of method found in MouseListener interface are given below:
- **public abstract void** actionPerformed(ActionEvent e);

MouseListener :

- The Java MouseListener is notified whenever you change the state of mouse.
- It is notified against MouseEvent.
- The MouseListener interface is found in java.awt.event package.
- It has five methods.
- The signature of 5 methods found in MouseListener interface are given below:

1. **public abstract void** mouseClicked(MouseEvent e);
2. **public abstract void** mouseEntered(MouseEvent e);
3. **public abstract void** mouseExited(MouseEvent e);
4. **public abstract void** mousePressed(MouseEvent e);
5. **public abstract void** mouseReleased(MouseEvent e);

KeyListener :

- The **Java KeyListener is notified whenever you change the state of key.**
- It is notified against KeyEvent.
- The KeyListener interface is found in java.awt.event package.
- it has three methods.
- The signature of 3 methods found in KeyListener interface are given below:

1. **public abstract void** keyPressed (KeyEvent e);
2. **public abstract void** keyReleased (KeyEvent e);
3. **public abstract void** keyTyped (KeyEvent e);

ItemListener :

- Item events are typically generated by components like checkboxes, radio buttons, and combo boxes when the user interacts with them, such as checking/unchecking a checkbox, selecting a radio button, or choosing an item from a combo box's list.
- To handle item events, need to implement the ItemListener interface. This interface defines a single method, itemStateChanged(ItemEvent e), which is called when an item event occurs.
- The signature of 1 method found in ItemListener interface are given below:

1. **public abstract void** itemStateChanged(ItemEvent e);

FocusListener :

- FocusListener is an interface in Java Swing that allows you to handle focus events on user interface components.
- Focus events occur when a component gains or loses focus, which can happen as a user interacts with a graphical user interface.
- For example, you might want to perform specific actions when a user clicks on a text field or when they move away from it.
- The signature of 2 method found in FocusListener interface are given below:

1. **public abstract void** focusGained(FocusEvent e);
2. **public abstract void** focusLost(FocusEvent e);