Experiment No. 11

Aim: Develop an application to demonstrate Event handling concept in GUI design.

Problem Statement:

1. Develop application to demonstrate Event handling in AWT GUI design.
2. Develop application to demonstrate Event handling in Swing GUI design.
3. Develop painting application in AWT using Mouse Event handling.

Theory:

Event:

- Event describes the change in state of any object.
- Events are generated as result of user interaction with the graphical user interface components.
- **Example :** Pressing a button, Entering a character in Textbox.

Event Handling-

1. It is the **mechanism** that controls the event and
2. Decides what should happen if an event occurs.
3. This mechanism has code which is known as **event handler.**
4. It is executed when an event occurs.

Event Handling Components:

- Events
- Events Source
- Listeners

Events :

- An event is a change in state of an object.

Events Source :

- Event source is an object that generates an event.
- It is an object on which event occurs.
- It is responsible for providing information of occurred event to it's handler.

Listeners :

- A listener is an object that listens to the event.
- It gets notified when an event occurs.
- It is responsible for generating response to an event
- It is also known as event handler.
- Listener waits until it receives an event.
- Once the event is received , the listener process the event and then returns.
- To design a listener class, have to develop some listener interfaces.

Java packages for Events

java.util, java.awt and java.awt.event.

Event Classes and Interfaces

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

Steps to handle events

- Implement appropriate interface in the class.
- Register the component with the listener.

Register the component with the Listener

Component	Listener
Button	public void addActionListener(ActionListener a){}
MenuItem	public void addActionListener(ActionListener a){}
TextField	public void addActionListener(ActionListener a){} public void addTextListener(TextListener a){}
TextArea	public void addTextListener(TextListener a){}
Checkbox	public void addItemListener(ItemListener a){}
Choice	public void addItemListener(ItemListener a){}
List	public void addActionListener(ActionListener a){} public void addItemListener(ItemListener a){}

Conclusion:

Students successfully studied Event and Event handling mechanism and developed various applications with event handling.