**8.8 Button :-**

This class creates a labeled button. The application can cause some action to happen when the button is pushed. This image depicts three views of a "Quit" button as it appears under the Solaris operating system:

The first view shows the button as it appears normally. The second view shows the button when it has input focus. Its outline is darkened to let the user know that it is an active object. The third view shows the button when the user clicks the mouse over the button, and thus requests that an action be performed.

The gesture of clicking on a button with the mouse is associated with one instance of ActionEvent, which is sent out when the mouse is both pressed and released over the button. If an application is interested in knowing when the button has been pressed but not released, as a separate gesture, it can specialize processMouseEvent, or it can register itself as a listener for mouse events by calling addMouseListener. Both of these methods are defined by Component, the abstract superclass of all components.

When a button is pressed and released, AWT sends an instance of ActionEvent to the button, by calling processEvent on the button. The button's processEvent method receives all events for the button; it passes an action event along by calling its own processActionEvent method. The latter method passes the action event on to any action listeners that have registered an interest in action events generated by this button.

If an application wants to perform some action based on a button being pressed and released, it should implement ActionListener and register the new listener to receive events from this button, by calling the button's addActionListener method. The application can make use of the button's action command as a messaging protocol.

**import** java.awt.\*;

**public** **class** ButtonExample {

**public** **static** **void** main(String[] args) {

    Frame f=**new** Frame("Button Example");

    Button b=**new** Button("Click Here");

    b.setBounds(50,100,80,30);

    f.add(b);

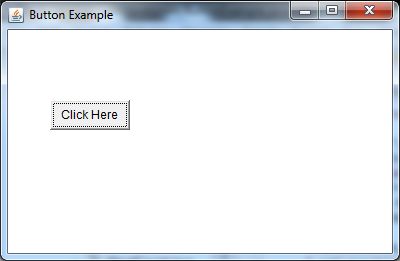
    f.setSize(400,400);

    f.setLayout(**null**);

    f.setVisible(**true**);

}

}



import java.awt.\*;  
    import java.applet.\*;  
    import java.awt.event.\*;

/\*

<HTML>  
<BODY>  
<APPLET ALIGN="CENTER" CODE="AppletSum.class" width = "700" height = "400"></APPLET>  
</BODY>  
</HTML>

\*/

    public class AppletSum extends Applet implements ActionListener{  
      TextField text1,text2,output;  
      Label label1,label2,label3;  
      Button button;  
      public void init(){  
        setLayout(null);  
        label1 = new Label("Enter Number1: ");  
        label1.setBounds(20,20,100,20);  
        add(label1);  
  
        text1 = new TextField(5);  
        text1.setBounds(150,20,100,20);  
        add(text1);  
  
        label2 = new Label("Enter Number2: ");  
        label2.setBounds(20,50,100,20);  
        add(label2);  
  
        text2 = new TextField(5);  
        text2.setBounds(150,50,100,20);  
        add(text2);  
  
        label3 = new Label("Sum of Two Numbers: ");  
        label3.setBounds(20,80,130,20);  
        add(label3);  
  
        output = new TextField(5);  
        output.setBounds(150,80,100,20);  
        add(output);  
  
        button = new Button("Sum");  
        button.setBounds(150,110,100,20);  
        add(button);  
  
        button.addActionListener(this);  
        }  
        public void actionPerformed(ActionEvent ae){  
        int num1=Integer.parseInt(text1.getText());  
        int num2=Integer.parseInt(text2.getText());  
        int sum=num1+num2;  
        output.setText(Integer.toString(sum));  
        }  
    }