Experiment no. 13

Date: 8/10/24

Roll no: B014

Aim: To develop GUI application using AWT controls and handling events.

Theory:

JAVA AWT (Abstract window Toolkit) is an API to develop graphical user interface or windows based applications in Java.

Java AWT components are platform dependent i.e. components are displayed according to the view of OS. AWT is heavy weight i.e. its components are using resources of underlying OS.

Event handlers:

'Action listener' in Java is an interface used for handling events triggered by user actions, like button clicks, its method 'actionperformed' contains the code to be executed when the event occurs.

Syntax:

```
button.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        // code to handle button click event
    }
});
```

Mouse Listener : 'Java MouseListener' is an interface in Java awt used for receiving mouse events like clicks, releases, enters, exits.

syntax:

```
public interface MouseListener
{
    void MouseClicked(MouseEvent e);
    void MousePressed(MouseEvent e);
    void MouseReleased(MouseEvent e);
    void MouseEntered(MouseEvent e);
    void MouseExited(MouseEvent e);
}
```

Mouse Motion Listener

'Java MouseMotionListener' is an interface in AWT used to handle mouse motion events, such as moving the mouse pointer or dragging it.

syntax:

```
public interface MouseMotionListener
{
    void mouseDragged(MouseEvent e);
    void mouseMoved(MouseEvent e);
}
```

4. KeyListener

Java KeyListener is an interface in AWT that provides methods to handle keyboard events, allowing

the program to respond to key presses and releases.

syntax:

```
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
class MyKeyListener implements KeyListener
{
    public void keyPressed(KeyEvent e)
    {
        // code to handle key pressed event.
    }
    public void keyReleased(KeyEvent e)
    {
        // code to handle key released event
    }
    public void keyTyped(KeyEvent e)
    {
        // code to handle key Typed event
    }
}
```

WindowListener ~~& WindowListener~~ interface in Java AWT is used to receive window events like opening, closing, minimising, maximising and resizing windows. It provides methods to handle these events.

syntax:

```
import java.awt.event.WindowListener;
public class YourClass implements WindowListener
```

```
{
    // implement windowlistener methods here
}
```

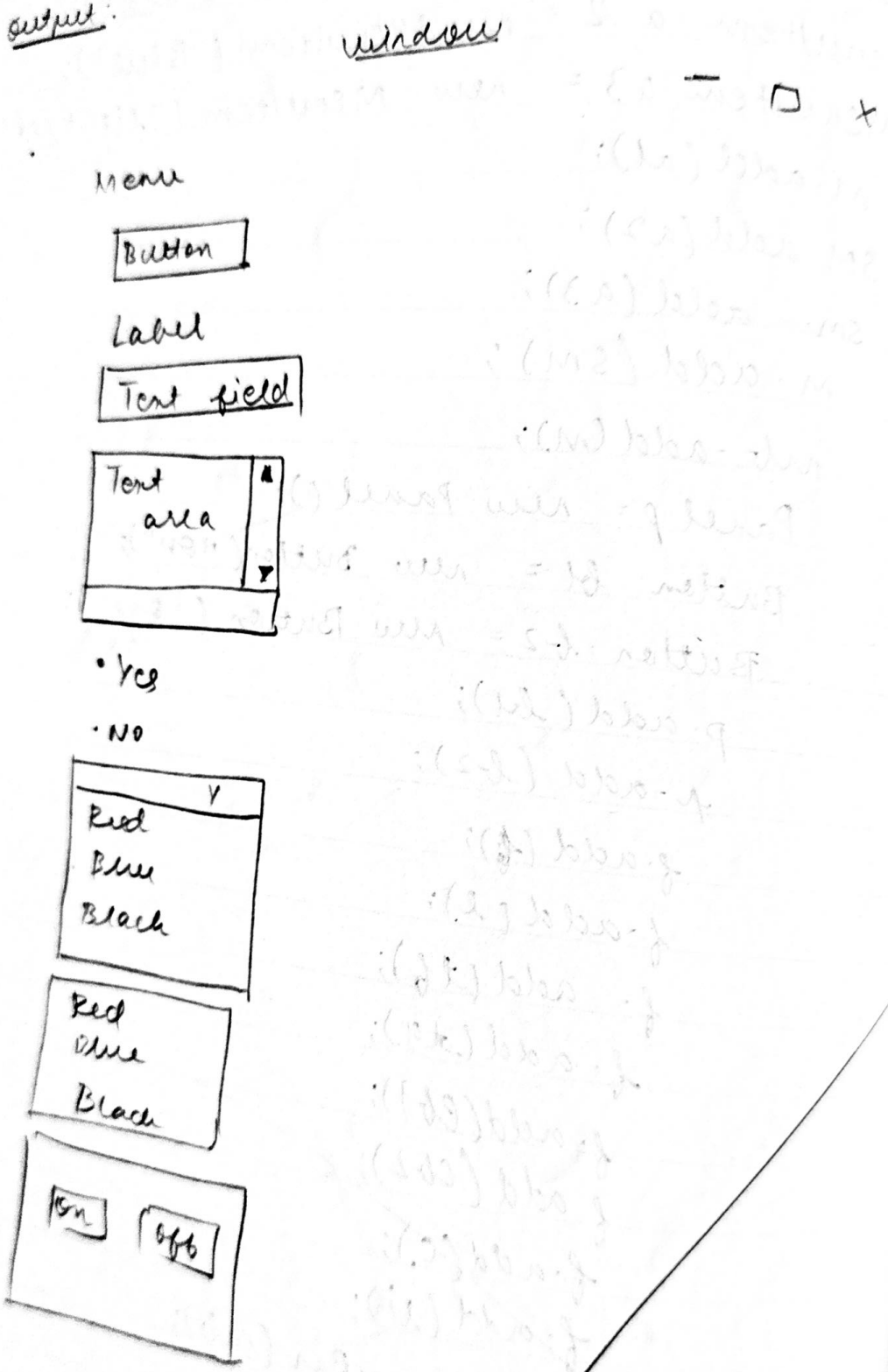
procedure:

program 1: (AWT controls)

```
import java.awt.*;
public class awtcontrols
{
    public static void main(String[] args)
    {
        Frame f = new Frame();
        Button b = new Button("Button");
        Label l = new label("label");
        Textfield tf = new Textfield("Textfield");
        Textarea ta = new Textarea("Text area");
        CheckBox group obj = new CheckBoxGroup();
        CheckBox cb1 = new CheckBox("Yes", obj, true);
        CheckBox cb2 = new CheckBox("NO", obj, false);
        Choice c = new Choice();
        c.add("Red");
        c.add("Blue");
        c.add("Black");
        List li = new list();
        li.add("Red");
        li.add("Blue");
        li.add("Black");
        MenuBar mb = new MenuBar();
```

```
Menu M = new Menu ("Menu");
MenuItem a1 = new MenuItem ("Red");
MenuItem a2 = new MenuItem ("Blue");
MenuItem a3 = new MenuItem ("Light red");
 M.add(a1);
SM.add(a2);
sm. aeld(a3);
 M.add (sM);
mb.add(M);
 Panel p = new Panel ();
 Button b1 = new Button("On");
 Button b2 = new Button ("Off");
  p.add(b1);
  p.add (b2);
   f.add(f);
   f.add(l);
   f. add(tf);
   f.add(ta);
    f.aeld(cb1);
    f.add(cb2);
    f.add(c);
    f.add(li);
    f.setMenuBar(mb);
     f.aeld(p);
    f.setsize(5000,3000);
    f.setLayout(null);
     f.setVisible(true);
 }
}
```

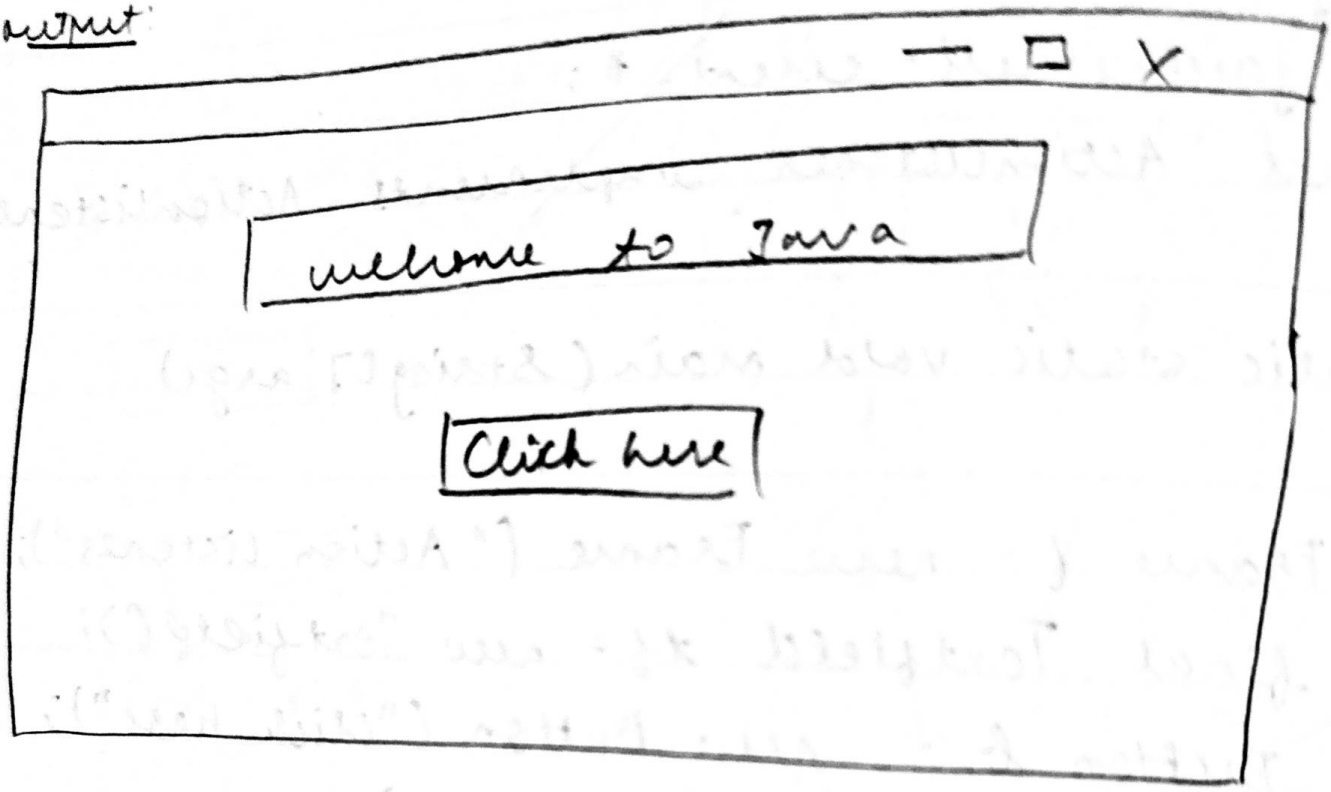
output:


window
Menu
Button
Label
Text field
Text area
Yes
No
Red
Blue
Black
Red
Blue
Black
On
Off

gram 2: (Action listener)

```
import java.awt.*;
import java.awt.event.*;
public class ActionListener implements ActionListener
{
    public static void main(String[] args)
    {
        Frame f = new Frame("Action Listener");
        final Textfield tf = new Textfield();
        Button b = new Button("Click here");
        b.addActionListener(this);
        f.add(b);
        f.add(tf);
        f.setSize(400,400);
        f.setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        tf.setText("welcome to Java");
    }
}
```

output



program 3: (Mouse Listener)

```
import java.awt.*;
import java.awt.event.*;
public class Mouselistener extends frame implements
mouselistener
{
    label l;
    Mouselistener Ex()
    {
        add.Mouselistener(this);
        l = new label();
        l.setBounds(20,50,100,20);
        add(l);
        setSize(300,300),
        setVisible(true);
    }
public void mouseclicked(MouseEnent e)
{
    l.setText("mouse clicked");
}
public void mouseEntered(MouseEnent e)
{
    l.setText("mouse entered");
}
public void mouseExited(MouseEnent e)
{
    l.setText("n mouse exited");
}
public void mousePressed(MouseEnent e)
{
```

output:

mouse entered

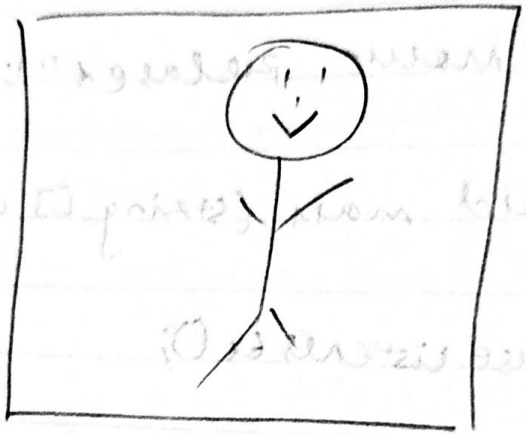
```
        l.setText("Mouse Pressed");
    }
    public void mouseReleased(MouseEvent e)
    {
        l.setText("Mouse released");
    }
    public static void main(String[] args)
    {
        new MouseListenerEx();
    }
}
```

Program 4: (Mouse Motion listener)

```
import java.awt.*;
import java.awt.event.*;
public class MouseMotionListEx extends Frame implements
MouseMotionListener
{
    MouseMotionListener()
    {
        addMouseMotionListener(this);
        setLayout(null);
        setVisible(true);
    }
    public void mouseDragged(MouseEvent e)
    {
        Graphics a = getGraphics();
        a.setColor(Color.Blue);
        a.fillOval(e.getX(), e.getY(), 20, 20);
    }
```

Ph:

Answer:

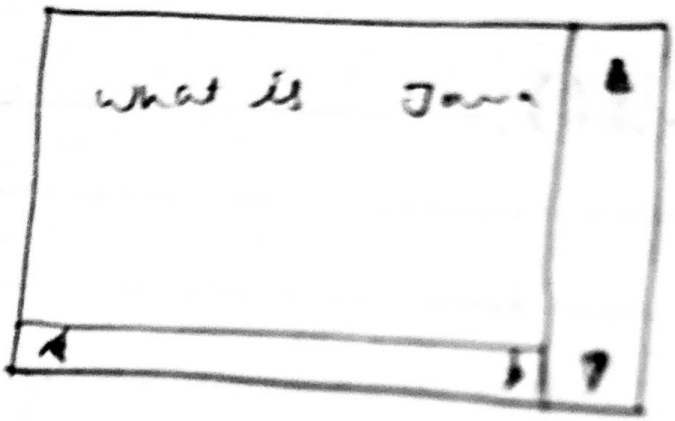


```
public void mouseMoved (MouseEvent e)
{
}
public static void main(String[] args)
 {
    new MouseMotionList();
 }
}
```

Program 5: (Keylistener)

```
import java.awt.*;
import java.awt.event.*;
public class Test extends Frame implements
KeyListener
{
    Label l;
    Textarea ta;
    Test()
    {
        l = new Label();
        ta = new TextArea();
        ta.addKeyListener(this);
        add(l);
        add(area);
        setVisible(true);
    }
    public void keyPressed(KeyEvent e)
    {
        l.setText("key pressed");
    }
```

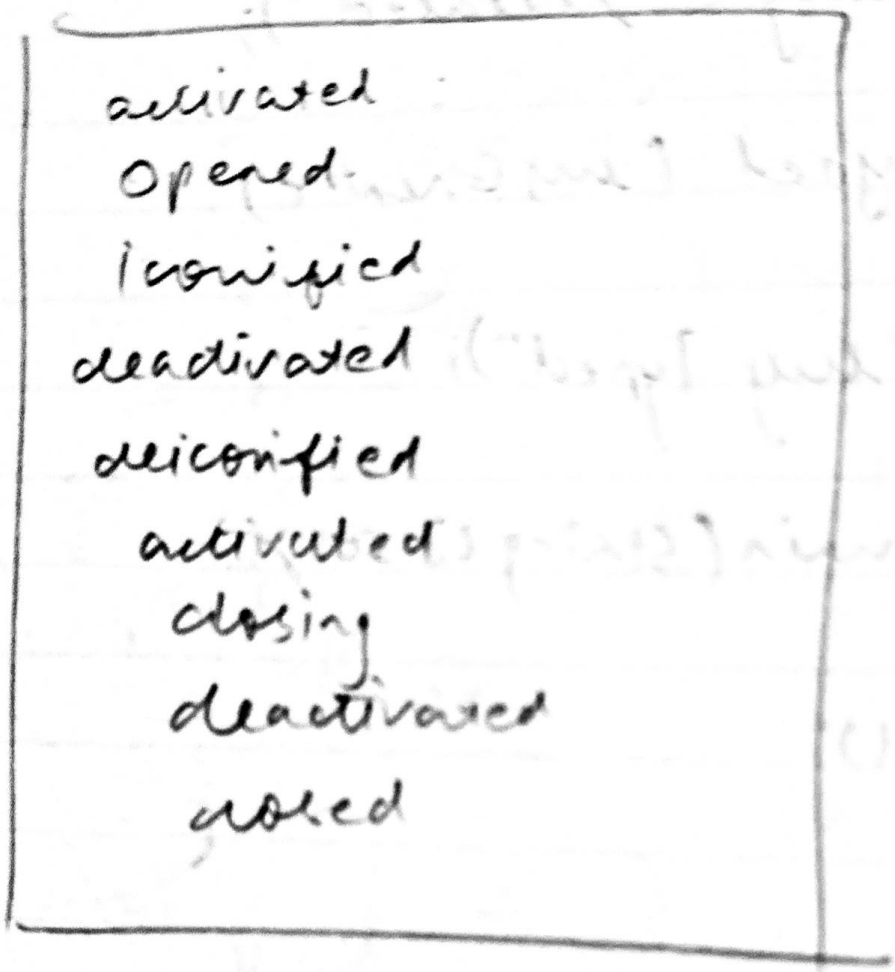
key pressed

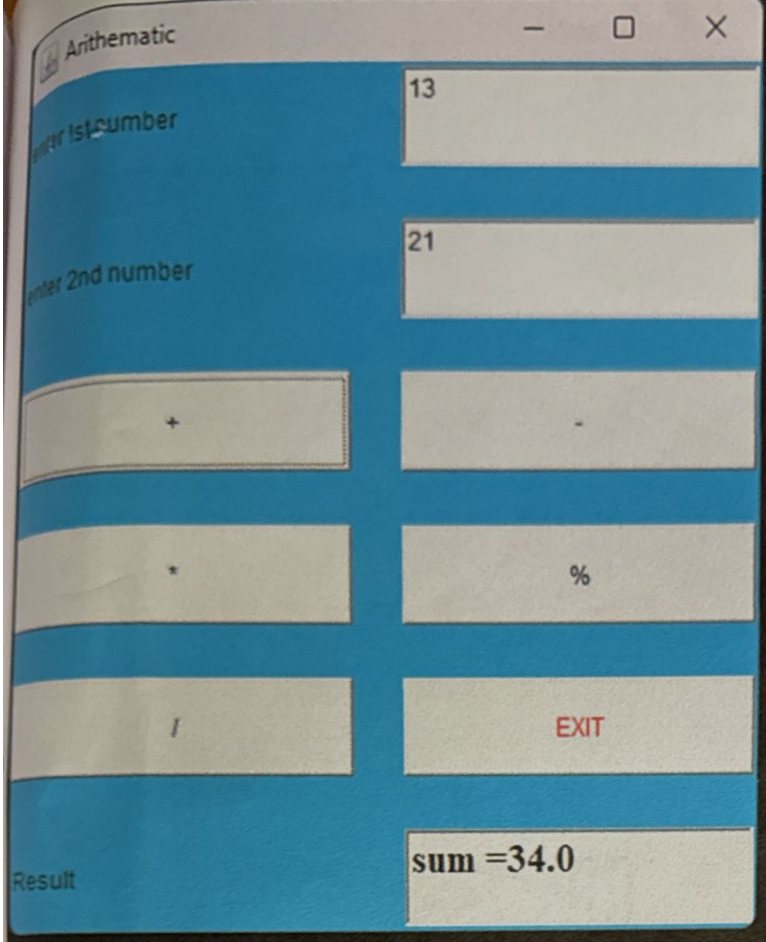


```
public void keyReleased (KeyEvent e)
  {
    l.setText ("key released");
  }
public void keyTyped (KeyEvent e)
  {
      l.setText ("key typed");
  }
public static void main(String [] args)
    {
        new Test();
    }
}
```

program6: (window listener)

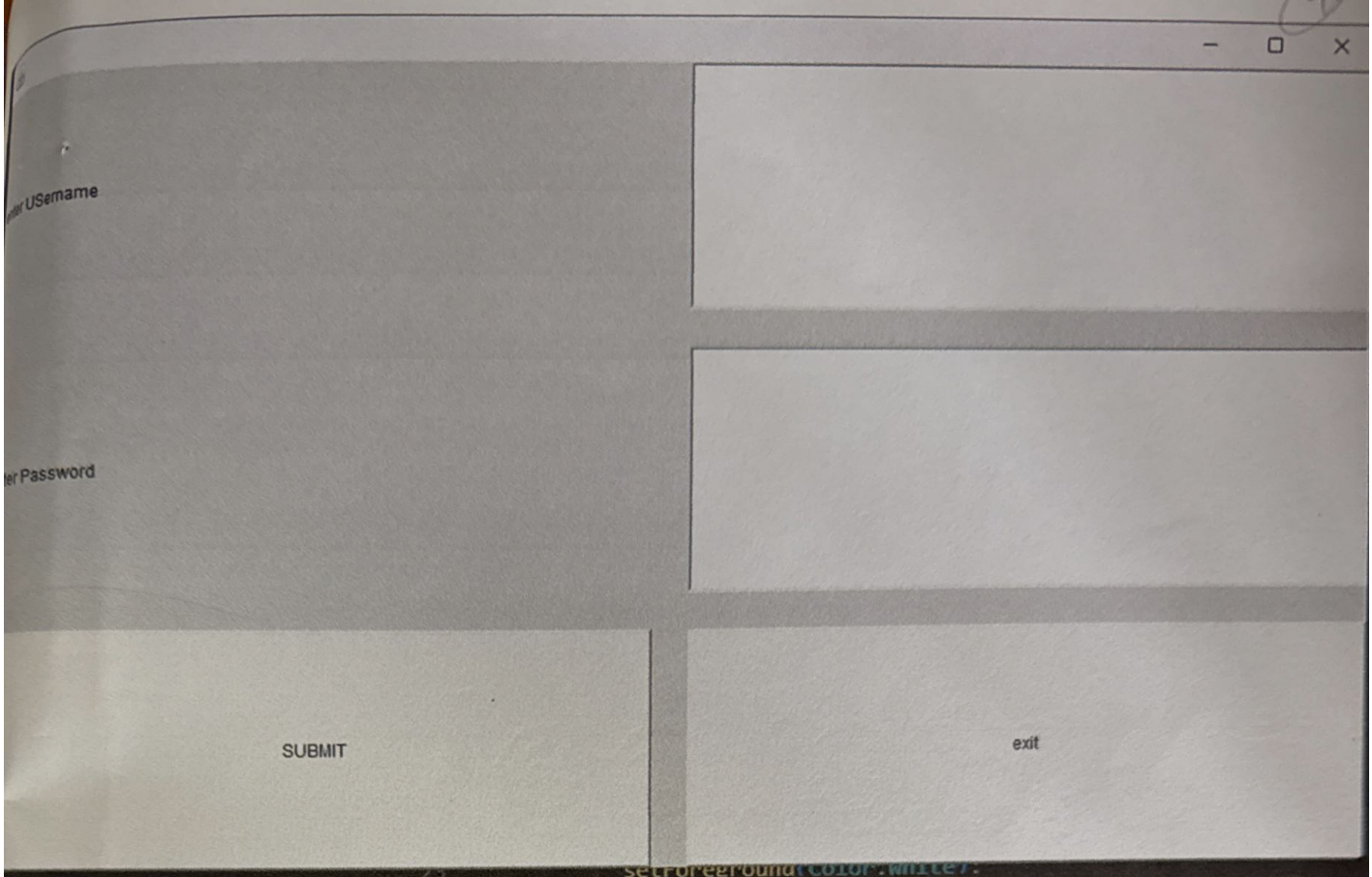
```
import java.awt.*;
import java.awt.event.*;
public class WindowExample extends Frame
implements WindowListener
 {
    WindowExample()
    {
      addWindowListener(this);
      setSize(400,400);
      setVisible(true);
    }
public static void main(String[] args)
  {
    new WindowExample();
  }
```


activated
Opened
iconified
deactivated
deiconified
activated
closing
deactivated
closed

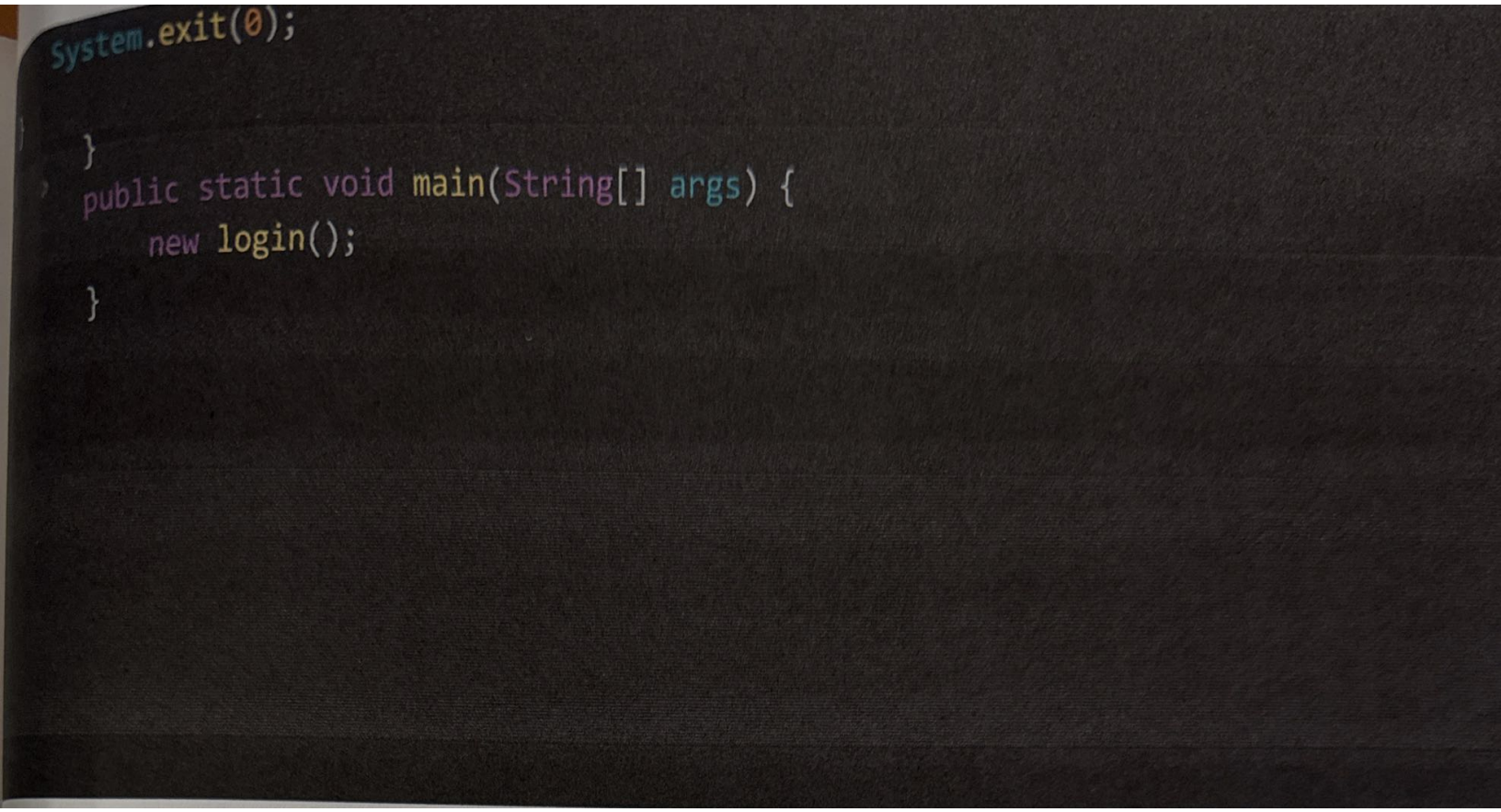



```
import java.awt.*;
import java.awt.event.*;
public class Calc extends Frame implements ActionListener {
   TextField tf1,tf2,tf3;
   Button pb,mb1,mulb2,rmb3,db,eb;
public Calc(){
        setLayout(new GridLayout(6,2,25,25));
        setBackground(Color.black);
        tf1=new TextField(10);
        tf2=new TextField(10);
        tf3=new TextField(10);
        pb=new Button("+");
        mb1=new Button("-");
        mulb2=new Button("*");
        rmb3=new Button("%");
        db=new Button("/");
        eb=new Button("EXIT");
        pb.addActionListener(this);
        mb1.addActionListener(this);
        mulb2.addActionListener(this);
        rmb3.addActionListener(this);
        db.addActionListener(this);
        eb.addActionListener(this);
        eb.setForeground(Color.RED);
        tf3.setEditable(false);
        tf3.setFont(new Font("serif",Font.BOLD,20));
        add(new Label("enter !st number"));
        add(tf1);
         add(new Label("enter 2nd number"));
        add(tf2);
        add(pb);
```

```
    add(mb1);
    add(mulb2);
    add(rmb3);
    add(db);
    add(eb);
     add(new Label("Result"));
    add(tf3);
    setTitle("Arithematic");
    setSize(400,450);
    setVisible(true);

blic void actionPerformed(ActionEvent e){
 Button btn=(Button)e.getSource();
 if(btn==eb){
    System.exit(0);
 }
 String s1=tf1.getText();
 double d1=Double.parseDouble(s1);
 double d2=Double.parseDouble(tf2.getText());
 String s2="";
 if(btn==pb){
     s2="sum ="+(d1+d2);
 }
 else if(btn==mb1){
     s2="difference ="+(d1-d2);
 }
 else if(btn==mulb2){
     s2="Product ="+(d1*d2);
 }
 else if(btn==rmb3){
     s2="remainder ="+(d1%d2);
 }
 else if(btn==db){
     s2="quotient ="+(d1/d2);

 }
         tf3.setText(s2);

}
Public static void main(String[] args) {
  new Calc();
}


}
```




```
ort java.awt.*;
ort java.awt.event.*;
lic class login extends Frame implements ActionListener {
 TextField tf1,tf2;
 Button sb,eb;
lic login(){
 setLayout(new GridLayout(3,2,25,25));
 setBackground(Color.LIGHT_GRAY);
 setVisible(true);
  tf1=new TextField(10);
      tf2=new TextField(10);
      sb=new Button("SUBMIT");
      eb=new Button("exit");
      add(new Label("enter USername"));
      add(tf1);
      add(new Label("enter Password"));
      add(tf2);
      tf2.setEchoChar('#');
      add(sb);
      add(eb);
      sb.addActionListener(this);
      eb.addActionListener(this);
      setForeground(Color.white);
      }
      public void actionPerformed(ActionEvent e){
 Button btn=(Button)e.getSource();
 if(btn==sb){
    tf2.setText("");
    tf1.setText("");
 }
```


System.exit(0);
}
public static void main(String[] args) {
new login();
}

```
public void windowActivated(WindowEvent args)
{
    System.out.println("activated");
}
public void windowClosed(WindowEvent args)
{
    System.out.println("closing");
}
public void windowDeactivated(WindowEvent args)
{
    System.out.println("deactivated");
}
public void windowDeiconified(WindowEvent args)
{
    System.out.println("deiconified");
}
public void windowOpened(WindowEvent args)
{
    System.out.println("opened");
}
```

Conclusion: Hence, we have developed GUI application using AWT controls.