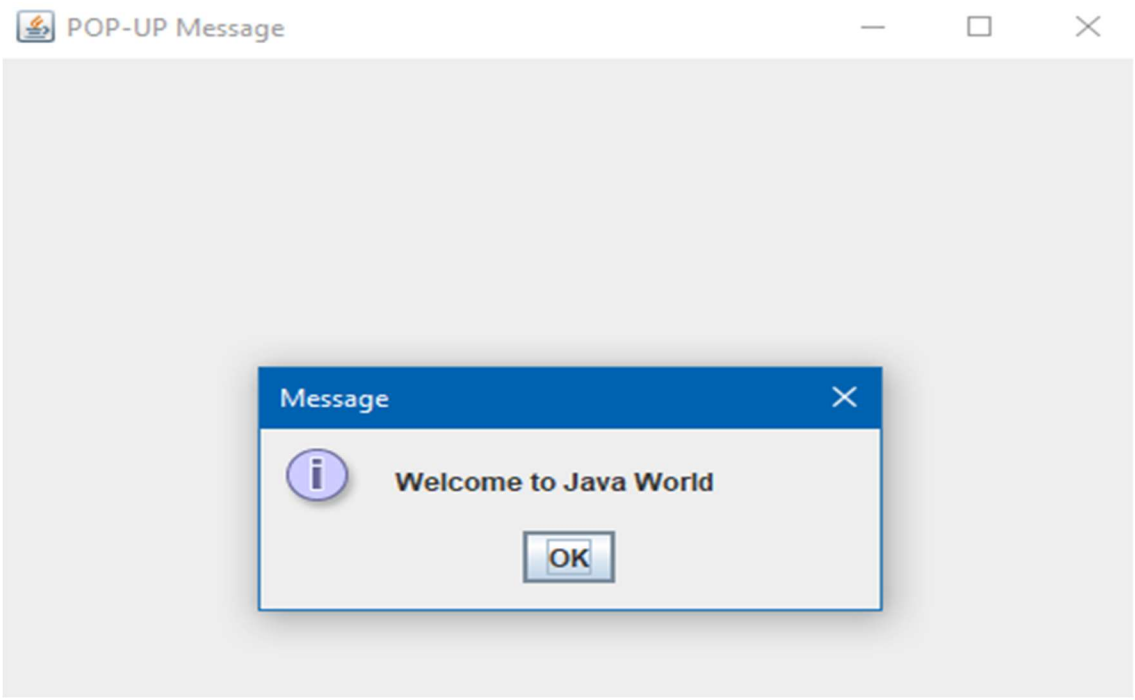
**Title**
**Create a GUI to show “Welcome to Java World” pop up message when GUI is launched.**

**Objective**
**To learn how to create a GUI that displays Welcome to Java World.**

**Program code:**

```
import javax.swing.*;
public class Main {
   public static void main(String[] args) {
            JFrame frame = new JFrame("POP-UP Message");
            frame.setSize(500,500);
            frame.setVisible(true);
            frame.setLayout(null);
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            JOptionPane.showMessageDialog(frame,"Welcome to Java
World");

   }
}
```

**OUTPUT:**



**Discussion**

**The practice for making a GUI that displays the message as passed in the program was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a GUI and display a pop up message.**

**Title**
**Create a GUI having two textbox that adds both values when button is clicked.**

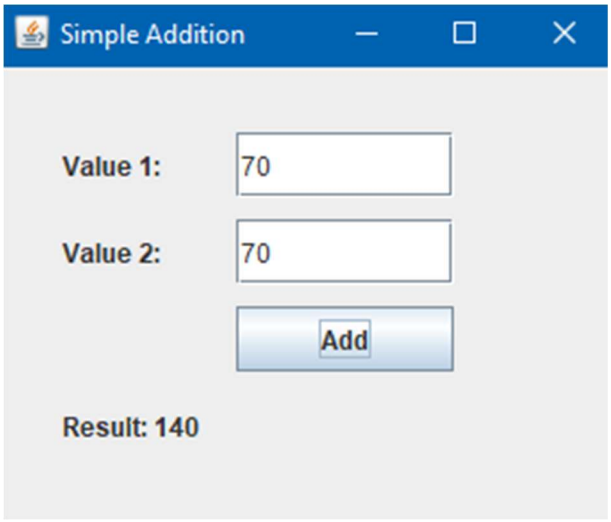
**Objective**
**To learn how to create a GUI with two textbox and a button which performs addition when button is clicked.**

**Program Code:**

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class Main {
   public static void main(String[] args) {
      JFrame f1 = new JFrame("Simple Addition");
      f1.setSize(300, 250);
      f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      f1.setLayout(null);
      f1.setVisible(true);
      JLabel labelnum1 = new JLabel("Value 1:");
      labelnum1.setBounds(30, 30, 80, 30);
      f1.add(labelnum1);
      JTextField num1 = new JTextField();
      num1.setBounds(110, 30, 100, 30);
      f1.add(num1);
      JLabel labelnum2 = new JLabel("Value 2:");
      labelnum2.setBounds(30, 70, 80, 30);
      f1.add(labelnum2);
      JTextField num2 = new JTextField();
      num2.setBounds(110, 70, 100, 30);
      f1.add(num2);
      JButton add = new JButton("Add");
      add.setBounds(110, 110, 100, 30);
      f1.add(add);
      JLabel result = new JLabel("Result:");
      result.setBounds(30, 150, 200, 30);
      f1.add(result);
      add.addActionListener(new ActionListener() {
         @Override
```

```
            public void actionPerformed(ActionEvent e) {
                try {
                    int val1 = Integer.parseInt(num1.getText());
                    int val2 = Integer.parseInt(num1.getText());
                    int sum = val1 + val2;
                    result.setText("Result: " + sum);
                } catch (NumberFormatException ex) {
                    result.setText("Please enter valid integers.");
                }
            }
        });
    }
}
```

**OUTPUT:**




**Discussion**

**The practice for making a GUI consisting two textbox which took inputs & performed addition on button click in the program was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a GUI with textbox and events and performing desired actions on click.**

**Title**
**Create a GUI having two buttons and swap the text when button is clicked.**

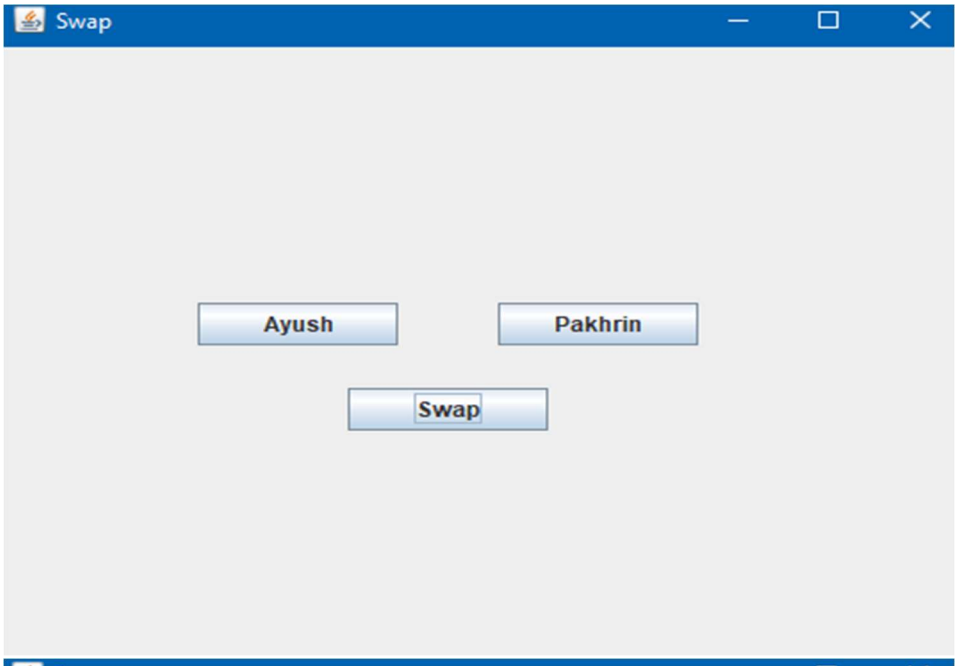
**Objective**
**To learn how to create a GUI with two buttons and button click swaps the value among the buttons on click.**

**Program Code:**

```
import javax.swing.*;
import java.awt.event.*;
public class Main {
   public static void main(String[] args) {
      JFrame f1=new JFrame("Swap");
      f1.setLayout(null);
      f1.setSize(500,500);
      f1.setVisible(true);
      f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      JButton b1=new JButton("Ayush");
      b1.setBounds(100,150,100,25);

      JButton b2=new JButton("Pakhrin");
      b2.setBounds(250,150,100,25);

      JButton swap=new JButton("Swap");
      swap.setBounds(175,200,100,25);
```

```
        f1.add(b1);
        f1.add(b2);
        f1.add(swap);

        swap.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String data1= b1.getText();
                String data2= b2.getText();
                String temp="";

                temp=data1;
                data1=data2;
                data2=temp;

                b1.setText(data1);
                b2.setText(data2);

            }
        });
    }
}
```

**OUTPUT:**





**Discussion**

**The practice for making a GUI having two buttons which swapped each other values onclick was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a GUI consisting buttons which swapped values onclick.**

**Title**

**Create a GUI of a calculator having four button(+,-,*,/) and perform related arithmetic operation on values when button is clicked.**

**Objective**

**To learn how to create a GUI of a calculator having all arithmetic operations performed on values when button is clicked.**

**Program Code:**

**Main.java**

```
public class Main {
    public static void main(String[] args) {
        Calculator calc=new Calculator();
    }
}
```

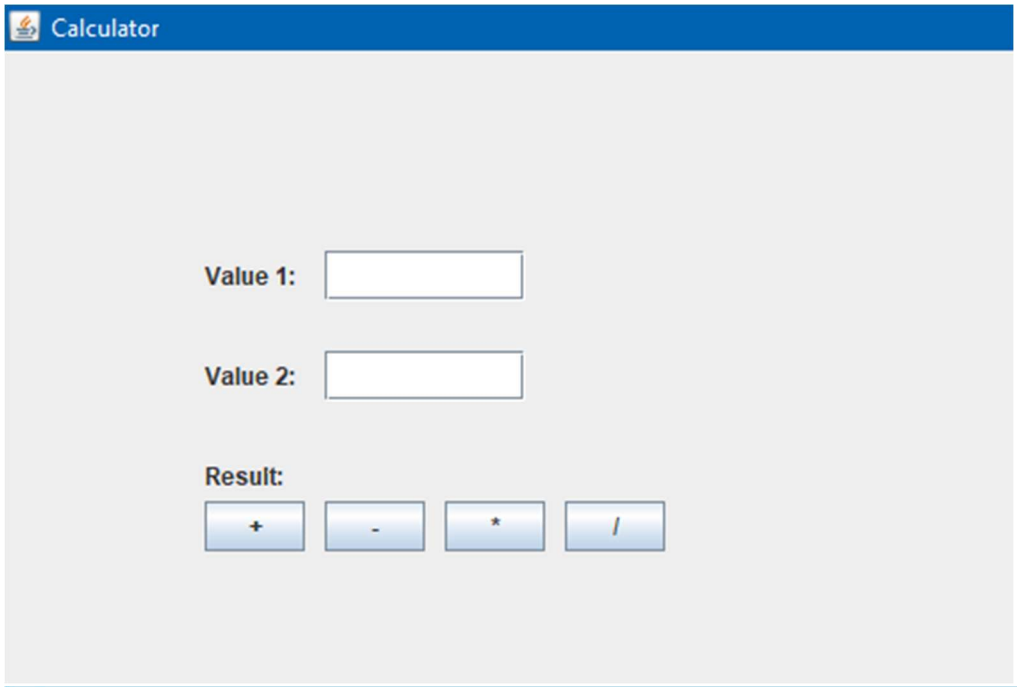
**Calculator.java**

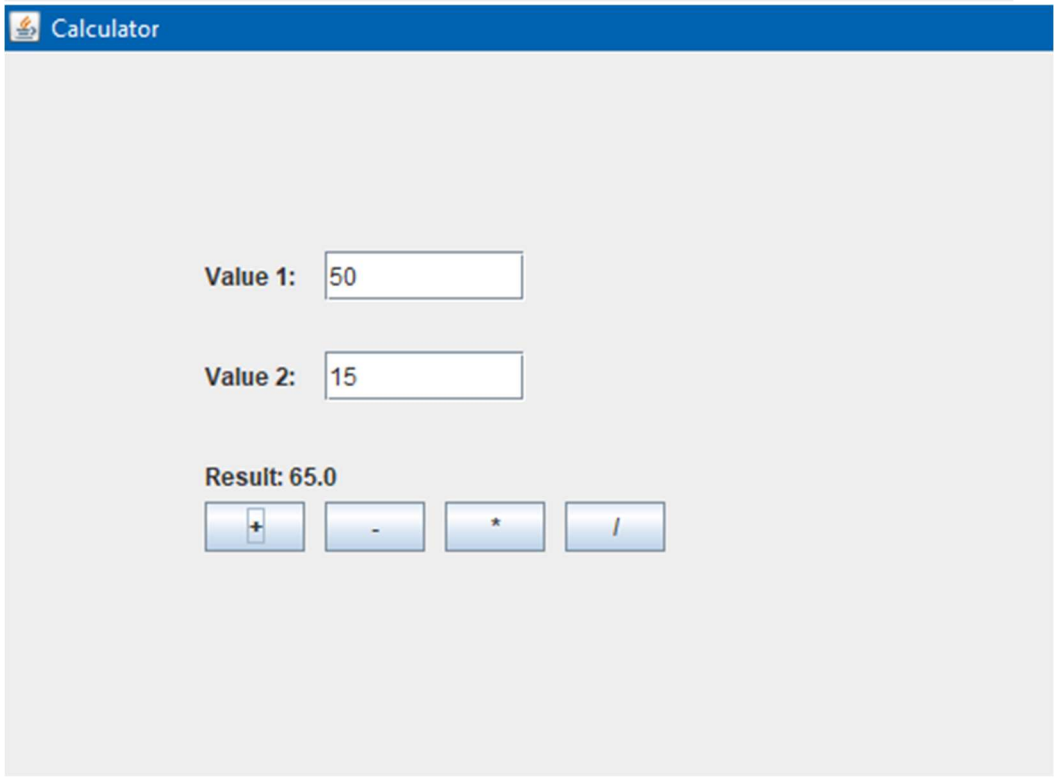
```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class Calculator implements ActionListener {
    public JFrame frame;
    public JLabel l1,l2,result;
    public JTextField val1,val2;
    public JButton add,sub,mul,div;


    public Calculator(){
        frame=new JFrame("Calculator");
        frame.setSize(500,500);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        l1=new JLabel("Value 1:");
        l1.setBounds(100,100,75,25);
```

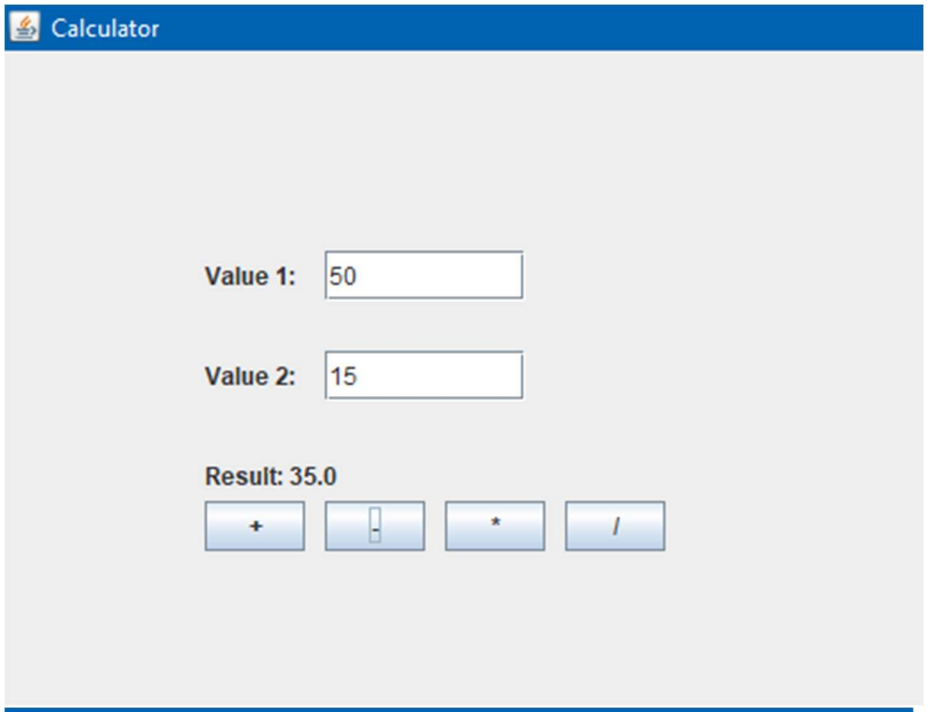
```
val1=new JTextField();
val1.setBounds(160,100,100,25);

l2=new JLabel("Value 2:");
l2.setBounds(100,150,75,25);

val2=new JTextField();
val2.setBounds(160,150,100,25);

result=new JLabel("Result:");
result.setBounds(100,200,100,25);

add=new JButton("+");
add.setBounds(100,225,50,25);

sub=new JButton("-");
sub.setBounds(160,225,50,25);

mul=new JButton("*");
mul.setBounds(220,225,50,25);

div=new JButton("/");
div.setBounds(280,225,50,25);

frame.add(l1);
frame.add(l2);
frame.add(val1);
frame.add(val2);
frame.add(result);
frame.add(add);
frame.add(sub);
frame.add(mul);
frame.add(div);

add.addActionListener(this);
sub.addActionListener(this);
mul.addActionListener(this);
```

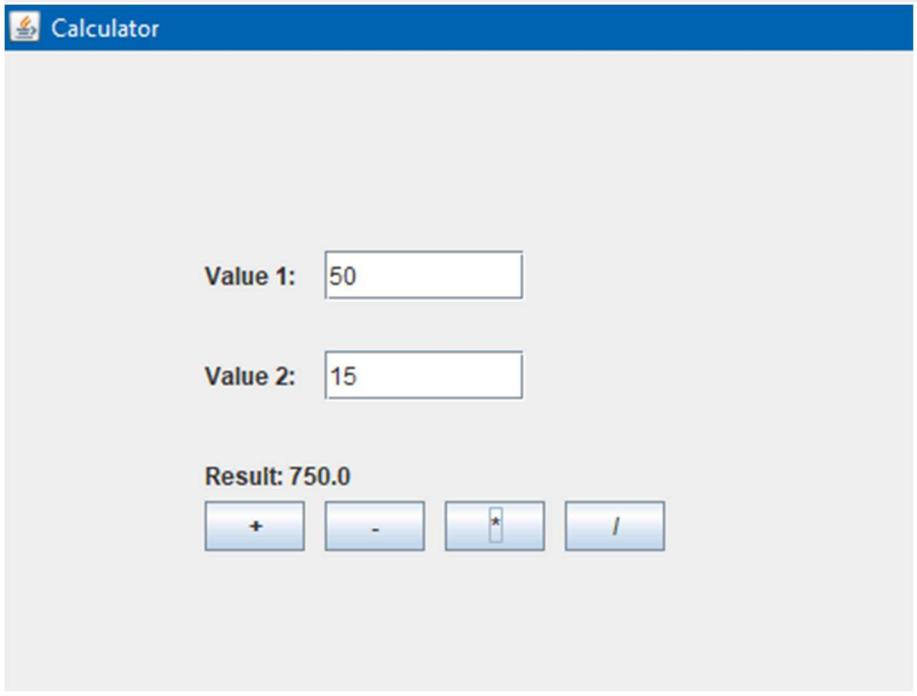
```
        div.addActionListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
    try {
        Double firstVal = Double.parseDouble(val1.getText());
        Double secondVal = Double.parseDouble(val2.getText());
        Double res=0.0;
        switch (e.getActionCommand()){
            case "+":
                res = firstVal + secondVal;
                break;
            case "-":
                res = firstVal - secondVal;
                break;
            case "*":
                res = firstVal * secondVal;
                break;
            case "/":
                if (secondVal != 0) {
                    res = firstVal / secondVal;
                } else {
                    JOptionPane.showMessageDialog(null, "Cannot divide by
zero");
                    return;
                }
                break;
        }
        result.setText("Result: " + res);
    }
    catch (Exception ex){
        System.out.println("Enter Valid Number!");
    }
    }

}
```

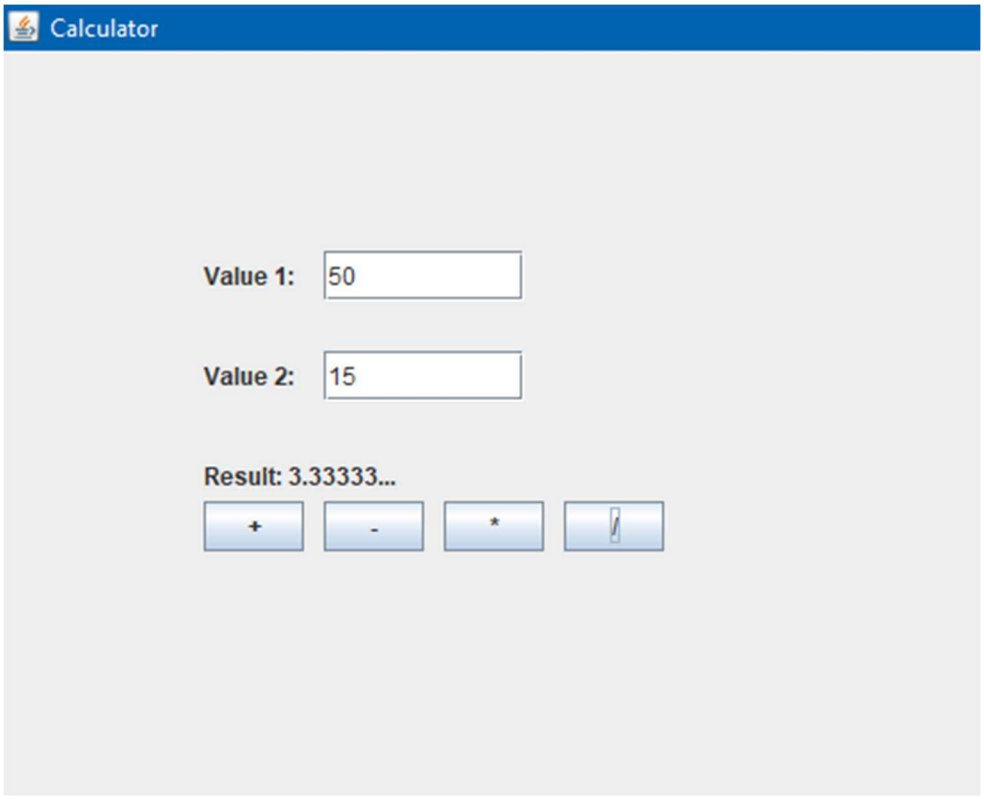
**OUTPUT:**


Calculator
Value 1:
Value 2:
Result:
+
-
*
/


Calculator
Value 1:
50
Value 2:
15
Result: 65.0
+
-
*
/


Calculator
Value 1:
50
Value 2:
15
Result: 35.0
+
-
*
/


Calculator
Value 1:
50
Value 2:
15
Result: 750.0
+
-
*
/




**Discussion**

**The practice for making a GUI of a calculator which performed arithmetic operations on values when button clicked was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a GUI of a calculator and performed desired arithmetic operations.**

**Title**

**Create a simple calculation that takes input in a box and display results in same box.**

**Objective**

**To learn how to create a GUI of a calculator having all input values in a box which is passed on click of input value buttons and arithmetic operations is performed on the button clicks and result displayed.**

**Program Code:**

**Main.java**

```
public class Main {
   public static void main(String[] args) {
      CalculatorGUI calc=new CalculatorGUI();
   }
}
```

**CalculatorGUI.java**

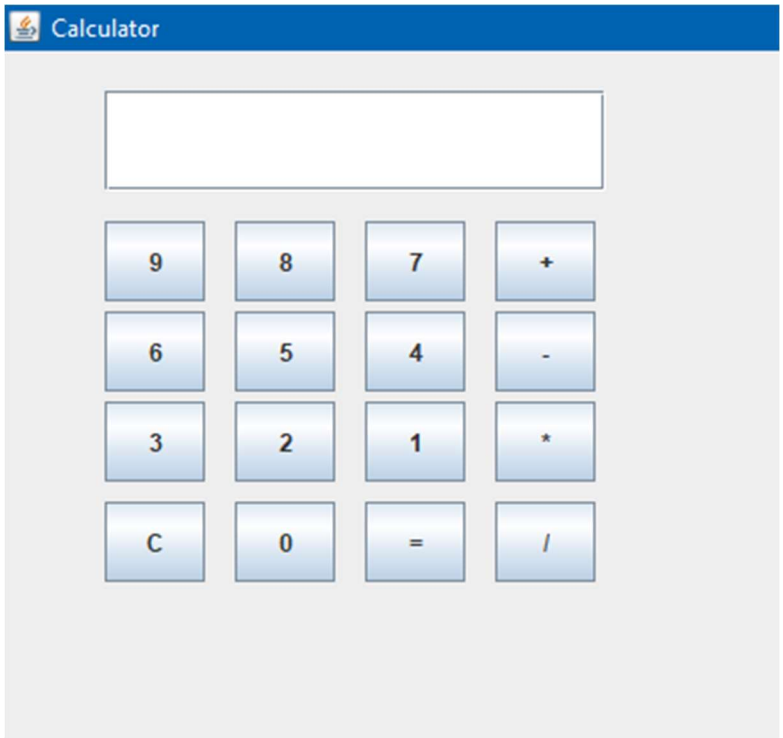
```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class CalculatorGUI implements ActionListener {
   JFrame f1;
   JTextField display;
   JButton b1,b2,b3,b4,b5,b6,b7,b8,b9,plus,minus,mul,div,zero,clear,res;
   double num1 = 0, num2 = 0, result = 0;
   char operator;

   public CalculatorGUI(){
      f1=new JFrame("Calculator");
      f1.setVisible(true);
      f1.setSize(600,800);
      f1.setLayout(null);
      f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      display=new JTextField();
```

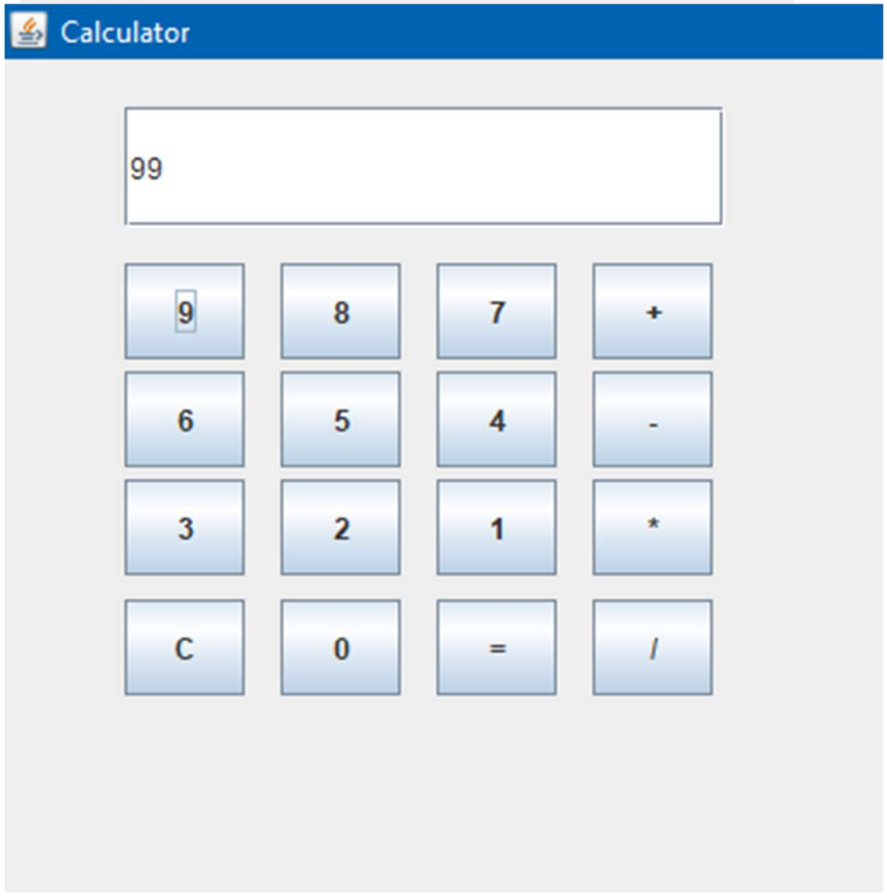
```
display.setBounds(50,20,250,50);

b1=new JButton("1");
b2=new JButton("2");
b3=new JButton("3");
b4=new JButton("4");
b5=new JButton("5");
b6=new JButton("6");
b7=new JButton("7");
b8=new JButton("8");
b9=new JButton("9");
plus=new JButton("+");
minus=new JButton("-");
mul=new JButton("*");
div=new JButton("/");
clear=new JButton("C");
zero=new JButton("0");
res=new JButton("=");

b1.setBounds(180,175,50,40);
b2.setBounds(115,175,50,40);
b3.setBounds(50,175,50,40);
b4.setBounds(180,130,50,40);
b5.setBounds(115,130,50,40);
b6.setBounds(50,130,50,40);
b7.setBounds(180,85,50,40);
b8.setBounds(115,85,50,40);
b9.setBounds(50,85,50,40);
plus.setBounds(245,85,50,40);
minus.setBounds(245,130,50,40);
mul.setBounds(245,175,50,40);
div.setBounds(245,225,50,40);
res.setBounds(180,225,50,40);
zero.setBounds(115,225,50,40);
clear.setBounds(50,225,50,40);

b1.addActionListener(this);
```

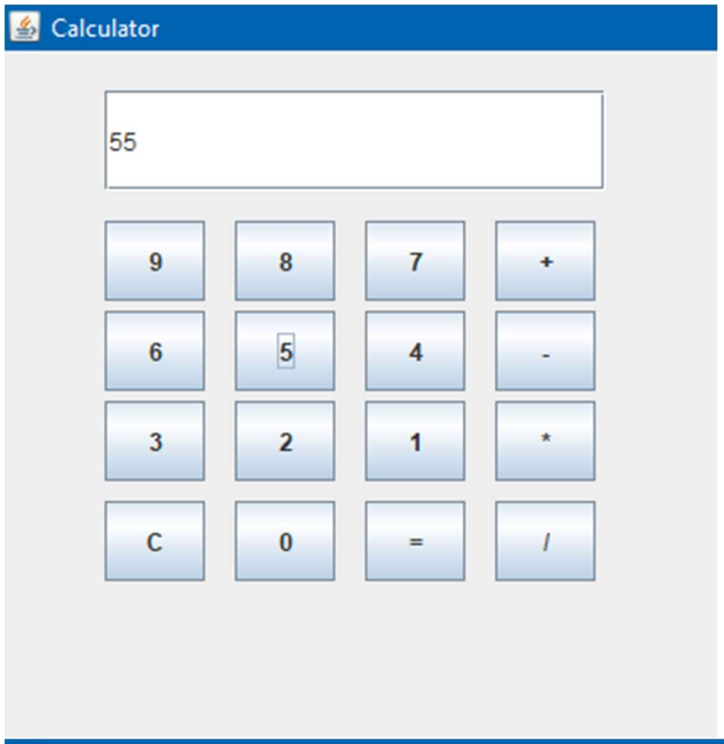
```
    b2.addActionListener(this);
    b3.addActionListener(this);
    b4.addActionListener(this);
    b5.addActionListener(this);
    b6.addActionListener(this);
    b7.addActionListener(this);
    b8.addActionListener(this);
    b9.addActionListener(this);
    plus.addActionListener(this);
    minus.addActionListener(this);
    mul.addActionListener(this);
    div.addActionListener(this);
    clear.addActionListener(this);
    res.addActionListener(this);
    zero.addActionListener(this);

    f1.add(b1);
    f1.add(b1);
    f1.add(b2);
    f1.add(b3);
    f1.add(b4);
    f1.add(b5);
    f1.add(b6);
    f1.add(b7);
    f1.add(b8);
    f1.add(b9);
    f1.add(plus);
    f1.add(minus);
    f1.add(mul);
    f1.add(div);
    f1.add(clear);
    f1.add(res);
    f1.add(zero);

    f1.add(display);
}
```

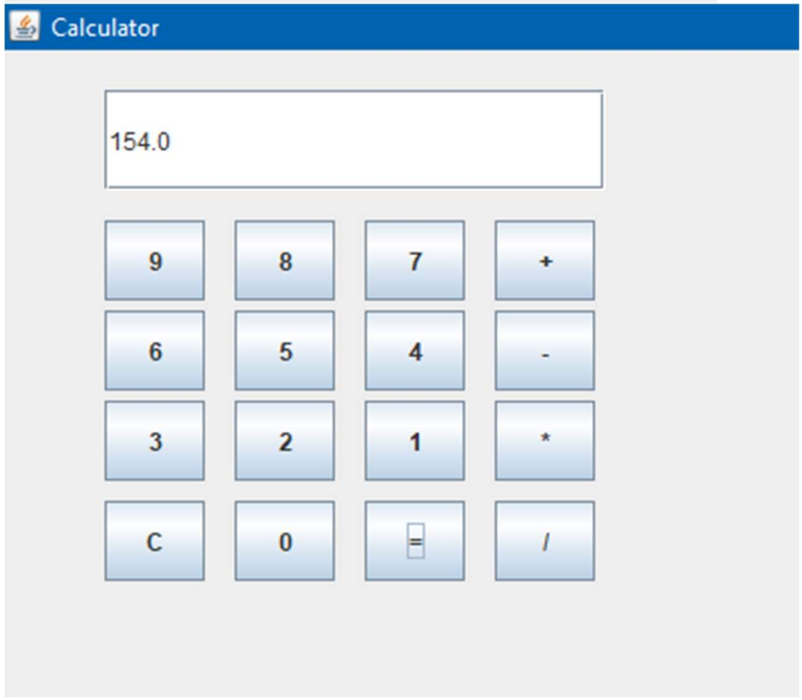
```
    @Override
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();
        if ((command.charAt(0) >= '0' && command.charAt(0) <= '9') ||
command.charAt(0) == '.') {
            display.setText(display.getText() + command);
        } else if (command.charAt(0) == 'C') {
            display.setText("");
            num1 = num2 = result = 0;
            operator = '\0';
        } else if (command.charAt(0) == '=') {
            num2 = Double.parseDouble(display.getText());
            switch (operator) {
                case '+':
                    result = num1 + num2;
                    break;
                case '-':
                    result = num1 - num2;
                    break;
                case '*':
                    result = num1 * num2;
                    break;
                case '/':
                    result = num1 / num2;
                    break;
            }
            display.setText(String.valueOf(result));

        } else {
            if (!display.getText().isEmpty()) {
                num1 = Double.parseDouble(display.getText());
                operator = command.charAt(0);
                display.setText("");
            }
        }

    }}
```

**OUTPUT:**




Calculator
55
9
8
7
+
6
5
4
-
3
2
1
*
C
0
=
/


Calculator
154.0
9
8
7
+
6
5
4
-
3
2
1
*
C
0
=
/

**Discussion**

**The practice for making a GUI of a calculator which performed arithmetic operations on values which are passed through input value buttons button clicked was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a GUI of a calculator which had all input values as button inputs and performed desired arithmetic operations.**

**Title**
**Create a gallery having some thumbnails with action listener and show full screen image when clicked.**

**Objective**
**To learn how to create a gallery of desired images thumbnails and on click the desired images to be displayed on full screen.**

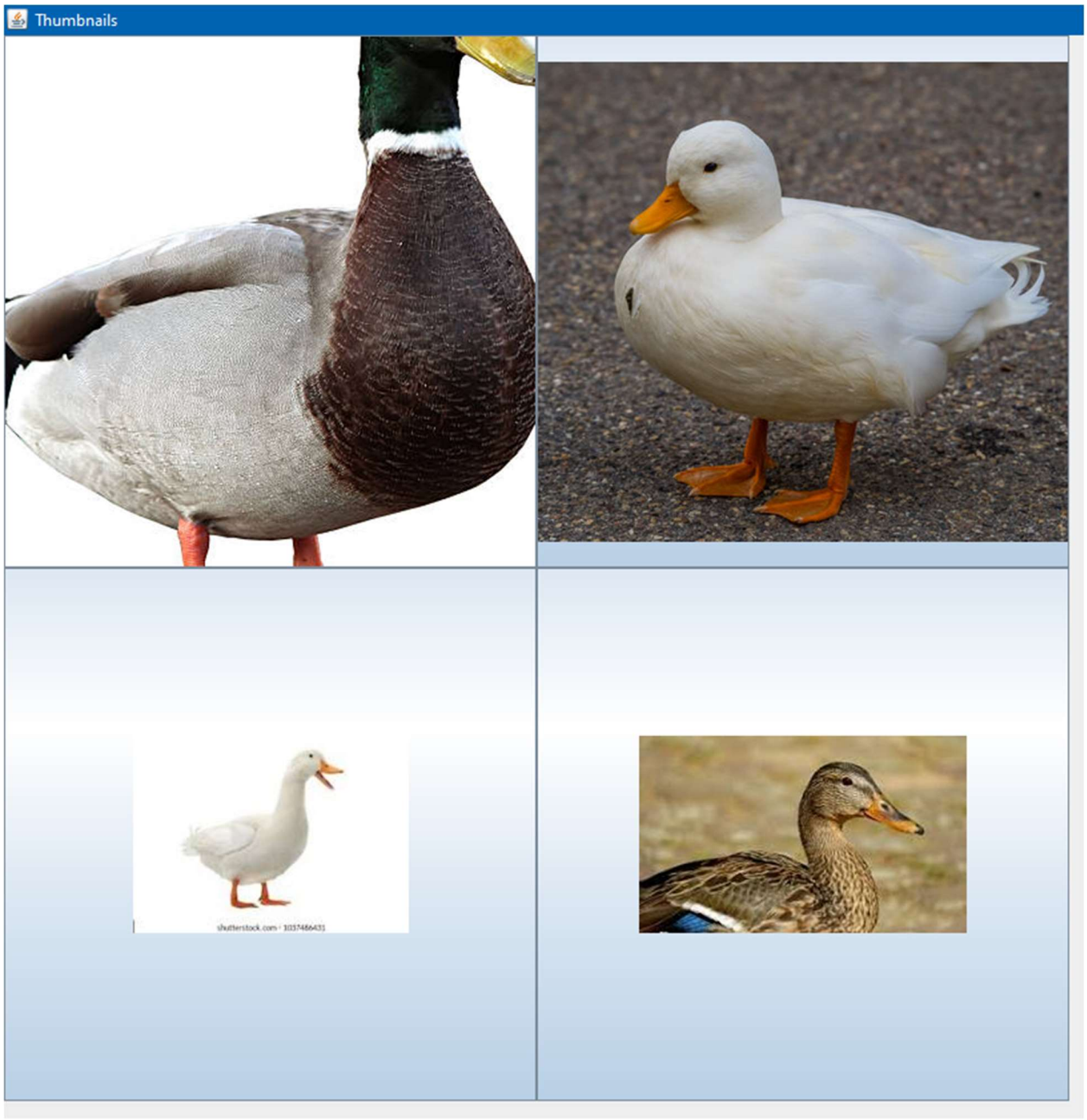
**Program Code:**

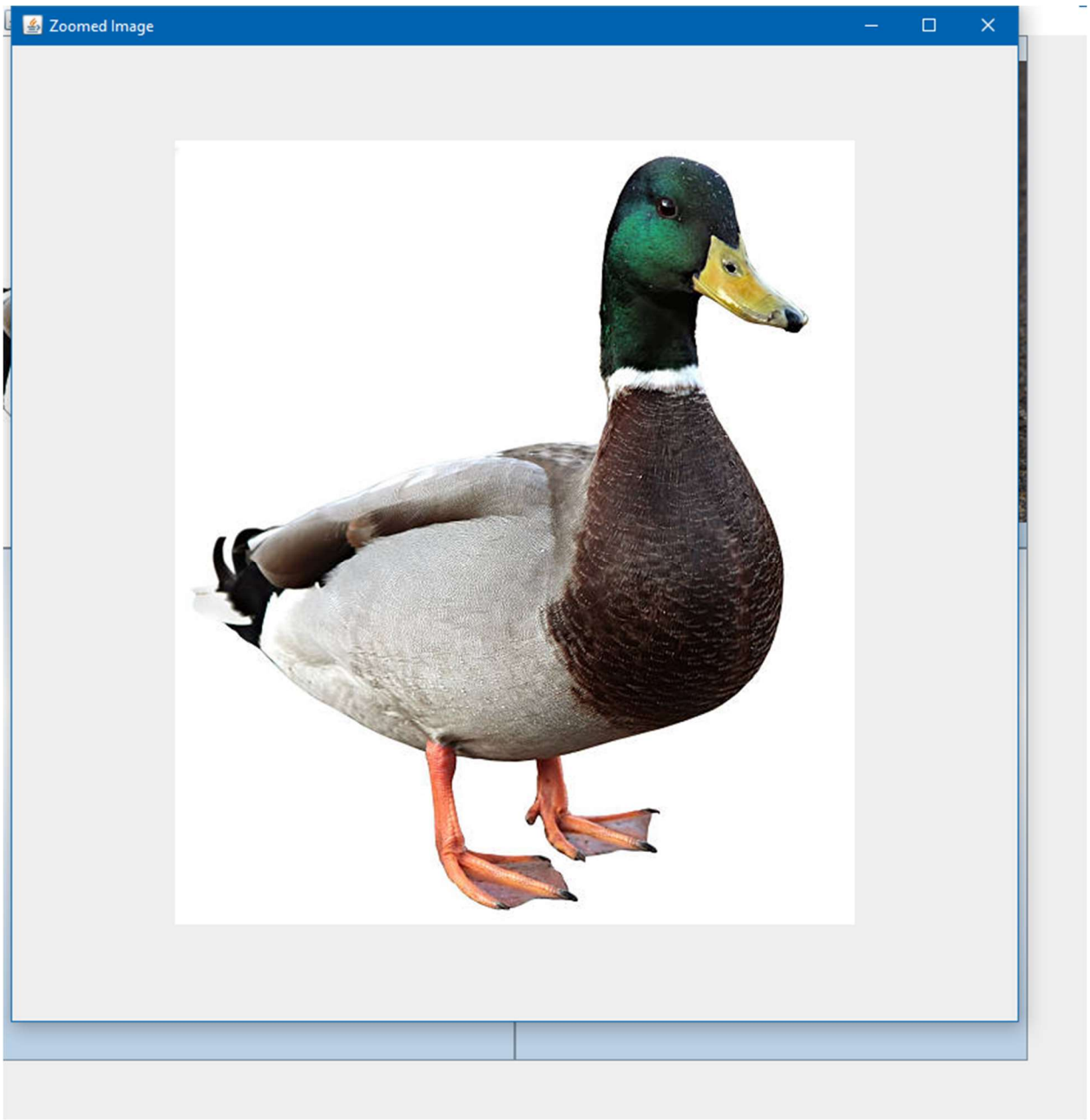
```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Main {
   public static void main(String[] args) {

       JFrame f1 = new JFrame("Thumbnails");
       f1.setLayout(null);
       f1.setSize(1000, 1000);
       f1.setVisible(true);
       f1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


       ImageIcon img1 = new ImageIcon("C:\\Users\\pakhr\\Desktop\\6th
sem\\AdvancedJava\\Lab reports\\lab6\\Images\\img1.jpg");
       ImageIcon img2 = new ImageIcon("C:\\Users\\pakhr\\Desktop\\6th
sem\\AdvancedJava\\Lab reports\\lab6\\Images\\img2.jpg");
       ImageIcon img3 = new ImageIcon("C:\\Users\\pakhr\\Desktop\\6th
sem\\AdvancedJava\\Lab reports\\lab6\\Images\\img3.jpg");
       ImageIcon img4 = new ImageIcon("C:\\Users\\pakhr\\Desktop\\6th
sem\\AdvancedJava\\Lab reports\\lab6\\Images\\img4.jpg");
       JButton b1 = new JButton(img1);
       JButton b2 = new JButton(img2);
       JButton b3 = new JButton(img3);
       JButton b4 = new JButton(img4);


       b1.setBounds(0, 0, 400, 400);
```

```
    b2.setBounds(400, 0, 400, 400);
    b3.setBounds(0, 400, 400, 400);
    b4.setBounds(400, 400, 400, 400);
    b1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            displayZoomedImage(img1);
        }
    });

    b2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            displayZoomedImage(img2);
        }
    });

    b3.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            displayZoomedImage(img3);
        }
    });

    b4.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            displayZoomedImage(img4);
        }
    });

    f1.add(b1);
    f1.add(b2);
    f1.add(b3);
    f1.add(b4);
}
public static void displayZoomedImage(ImageIcon icon) {
    JFrame f2 = new JFrame("Zoomed Image");
    f2.setSize(800, 800);
    JLabel l2 = new JLabel(icon);
    f2.add(l2);
```

```
        f2.setVisible(true);
    }
}
```

**OUTPUT:**






**Discussion**

**The practice for making a GUI of a gallery of desired images thumbnails and on click the desired images to be displayed on full screen with the help of action listener was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a gallery of images and display desire image on full screen via the help of action listener.**