Topics to cover:

1. Swing classes

1. Swing classes

There are plenty of classes in Swing to work with. We will discuss some of the most important ones.

1.1 JButton class

The JButton class is used to create a labeled button that has platform independent implementation. The application results in some action when the button is pushed. It inherits the AbstractButton class.

1.1.1 Commonly used Constructors for JButton class:

The most commonly constructors are show in the table below.

Constructor	Description
JButton()	It creates a button with no text and icon.
JButton(String s)	It creates a button with the specified text.

JButton(Icon i)	It creates a button with the specified icon object.

Table 1: The most commonly used constructors in JButton class

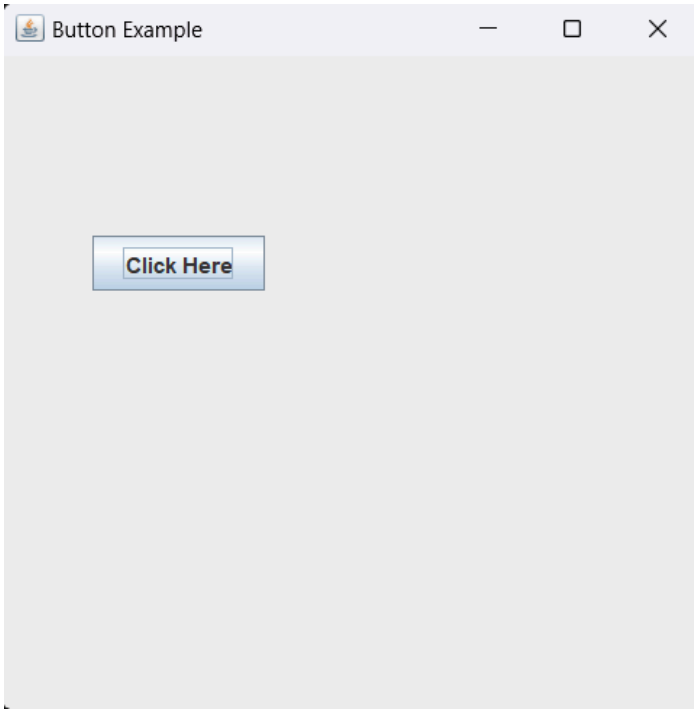
1.1.2 Java JButton Example

Program1:

```
import javax.swing.*;

public class JButton1 {
public static void main(String[] args) {
    JFrame f=new JFrame("Button Example");
    JButton b=new JButton("Click Here");
    b.setBounds(50,100,95,30);
    f.add(b);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
}
}
```

Output:



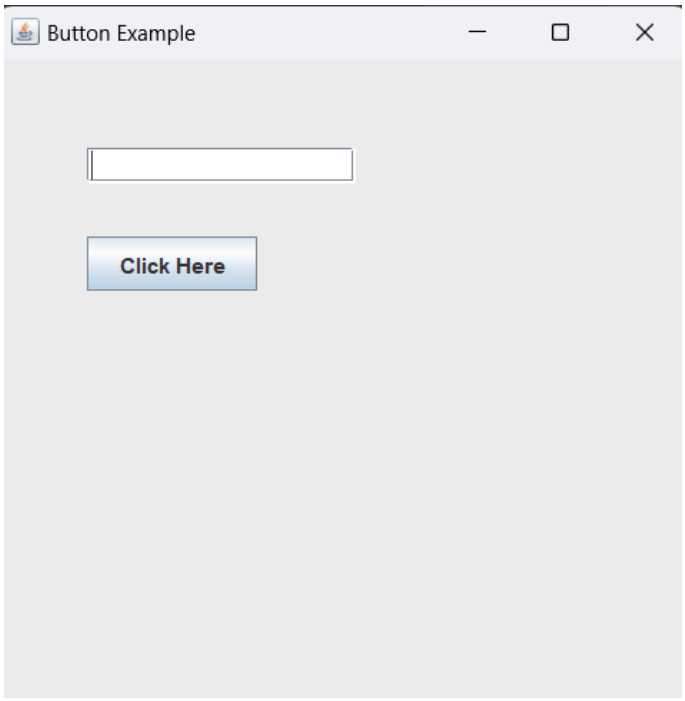
1.1.4 Java JButton Example with ActionListener

Program1:

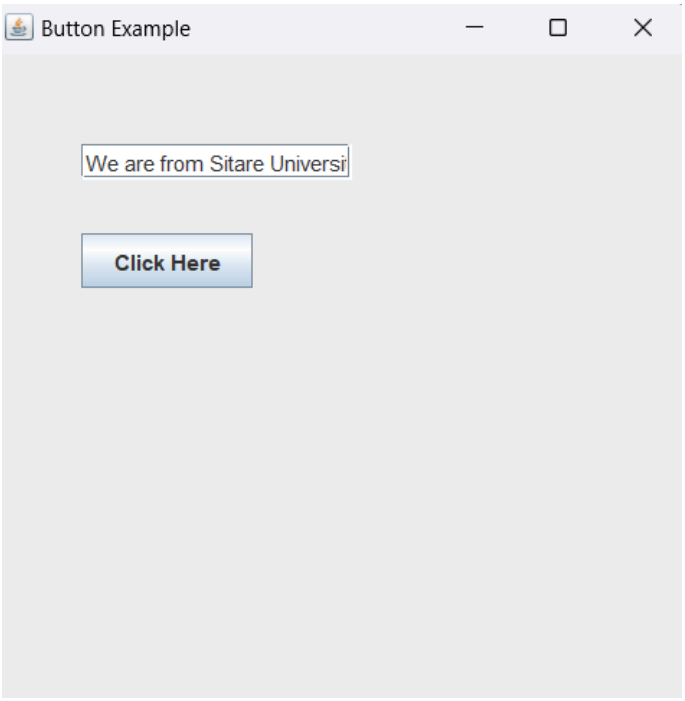
```
import java.awt.event.*;
import javax.swing.*;
public class ButtonExample {
public static void main(String[] args) {
    JFrame f=new JFrame("Button Example");
    final JTextField tf=new JTextField();
    tf.setBounds(50,50, 150,20);
    JButton b=new JButton("Click Here");
    b.setBounds(50,100,95,30);
    b.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent e){
            tf.setText("We are from Sitare University.");
        }
    });
```

```
    f.add(b);f.add(tf);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
}
}
```

Output:



In the above output with textfield, if you click on the button then you will see the output as follows.



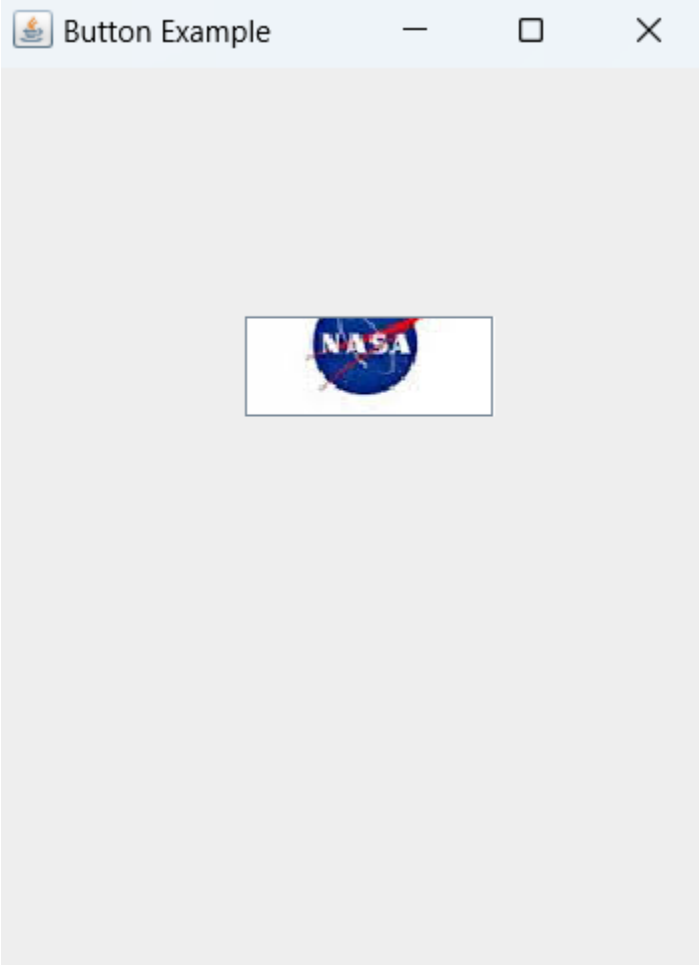
1.1.5 Example of displaying image on the button:

Program1:

```
import javax.swing.*;

public class JButtonImage{
JButtonImage(){
JFrame f=new JFrame("Button Example");
JButton b=new JButton(new ImageIcon("C:\\Users\\Aman\\Pictures\\logo.png"));
b.setBounds(100,100,100, 40);
f.add(b);
f.setSize(300,400);
f.setLayout(null);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
public static void main(String[] args) {
    new JButtonImage();
}
}
```

Output:



1.2 JLabel class

The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

1.2.1 Commonly used Constructors for JLabel class:

The most commonly used constructors are shown in the table below.

Constructor	Description
JLabel()	Creates a JLabel instance with no image and with an empty string for the title.
JLabel(String s)	Creates a JLabel instance with the specified text.
JLabel(Icon i)	Creates a JLabel instance with the specified image.
JLabel(String s, Icon i, int horizontalAlignment)	Creates a JLabel instance with the specified text, image, and horizontal alignment.

Table 3: The most commonly used constructors in JLabel class

1.2.2 Commonly used Methods of JButton class:

The most commonly used methods are shown in the table below.

Methods	Description
String getText()	t returns the text string that a label displays.
void setText(String text)	It defines the single line of text this component will display.
void setHorizontalAlignment(int alignment)	It sets the alignment of the label's contents along the X axis.
Icon getIcon()	It returns the graphic image that the label displays.
int getHorizontalAlignment()	It returns the alignment of the label's contents along the X axis.

Table 4: The most commonly used methods

1.2.3 Java JLabel Example

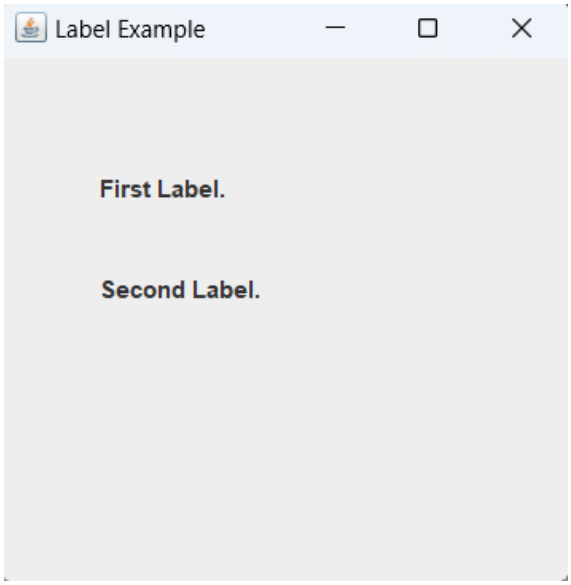
Program1:

```
import javax.swing.*;

class JLabel1
{
public static void main(String args[])
   {
   JFrame f= new JFrame("Label Example");
   JLabel l1,l2;
   l1=new JLabel("First Label.");
   l1.setBounds(50,50, 100,30);
```

```
l2=new JLabel("Second Label.");
l2.setBounds(50,100, 100,30);
f.add(l1); f.add(l2);
f.setSize(300,300);
f.setLayout(null);
f.setVisible(true);
}
}
```

Output:



1.2.4 Java JLabel Example with ActionListener

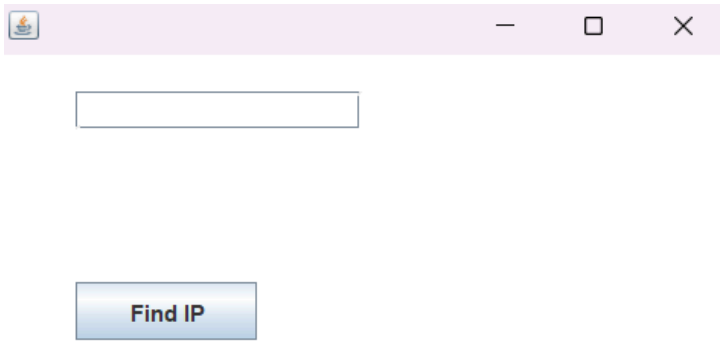
Program1:

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

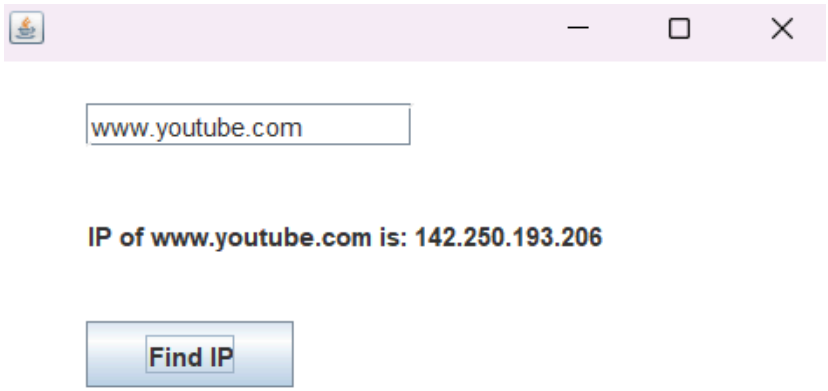
```
public class JLabel2 extends Frame implements ActionListener{
   JTextField tf; JLabel l; JButton b;
   JLabel2(){
      tf=new JTextField();
      tf.setBounds(50,50, 150,20);
      l=new JLabel();
      l.setBounds(50,100, 250,20);
      b=new JButton("Find IP");
      b.setBounds(50,150,95,30);
      b.addActionListener(this);
      add(b);add(tf);add(l);
      setSize(400,400);
      setLayout(null);
      setVisible(true);
   }
   public void actionPerformed(ActionEvent e) {
      try{
      String host=tf.getText();
      String ip=java.net.InetAddress.getByName(host).getHostAddress();
      l.setText("IP of "+host+" is: "+ip);
      }catch(Exception ex){System.out.println(ex);}
   }
   public static void main(String[] args) {
      new JLabel2();
   } }
```

Output:

At first, the screen will show the following output.



After that, if we type a particular website such as “www.youtube.com” and then click on “Find IP” button, this will show the IP address of this website as follows.



1.3 JTextField class

The object of a JTextField class is a text component that allows the editing of a single line text. It inherits the JTextComponent class.

1.3.1 Commonly used Constructors:

The most commonly used constructors are shown in the table below.

Constructor	Description
JTextField()	Creates a new TextField
JTextField(String text)	Creates a new TextField initialized with the specified text.
JTextField(String text, int columns)	Creates a new TextField initialized with the specified text and columns.
JTextField(int columns)	Creates a new empty TextField with the specified number of columns.

Table 5: The most commonly used constructors

1.3.2 Commonly used Methods:

The most commonly used methods are shown in the table below.

Methods	Description

void addActionListener(ActionListener l)	It is used to add the specified action listener to receive action events from this textfield.
Action getAction()	It returns the currently set Action for this ActionEvent source, or null if no Action is set.
void setFont(Font f)	It is used to set the current font.
void removeActionListener(ActionListener l)	It is used to remove the specified action listener so that it no longer receives action events from this textfield.

Table 6: The most commonly used methods

1.3.4 Java JTextField Example

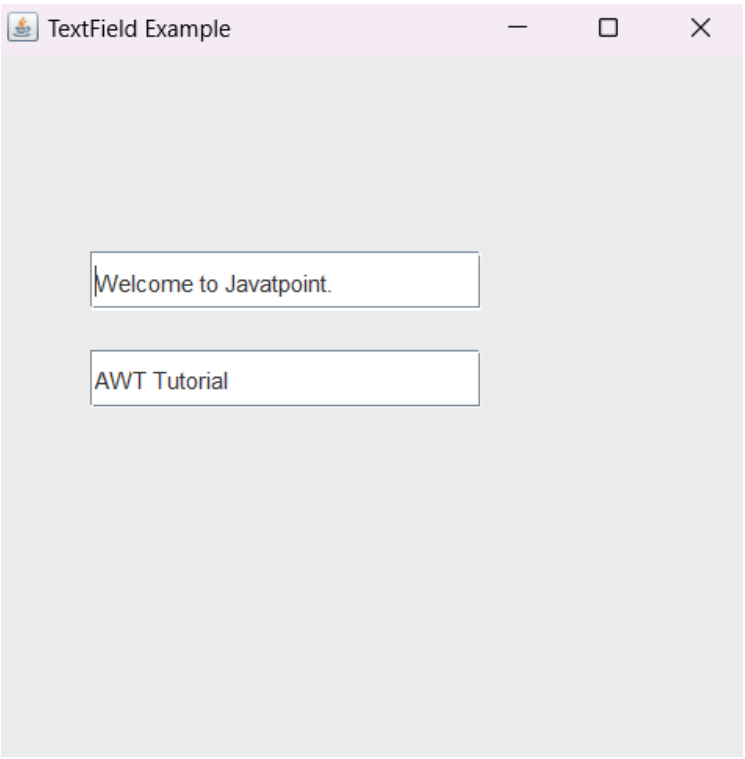
Program1:

```
import javax.swing.*;

class TextFieldExample
{
public static void main(String args[])
    {
    JFrame f= new JFrame("TextField Example");
    JTextField t1,t2;
    t1=new JTextField("Welcome to Javatpoint.");
    t1.setBounds(50,100, 200,30);
    t2=new JTextField("AWT Tutorial");
```

```
t2.setBounds(50,150, 200,30);
f.add(t1); f.add(t2);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
}
}
```

Output:



1.3.5 Java JTextField Example with ActionListener

Program1:

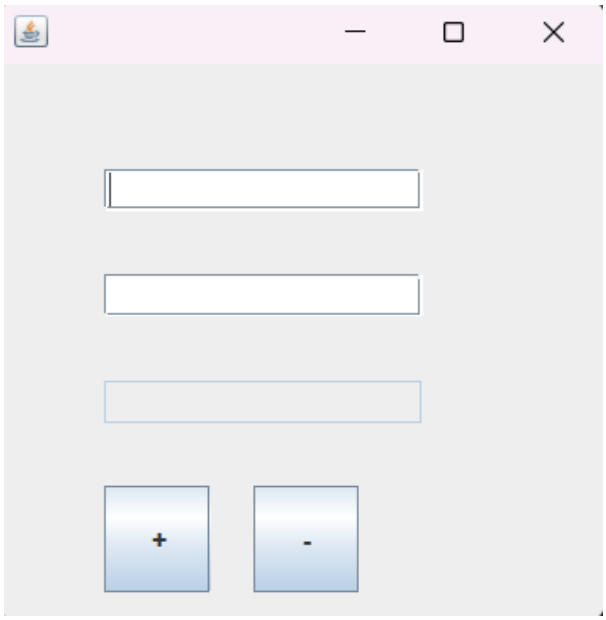
```
import javax.swing.*;
import java.awt.event.*;
```

```
public class TextFieldExample implements ActionListener{
    JTextField tf1,tf2,tf3;
    JButton b1,b2;
    TextFieldExample(){
        JFrame f= new JFrame();
        tf1=new JTextField();
        tf1.setBounds(50,50,150,20);
        tf2=new JTextField();
        tf2.setBounds(50,100,150,20);
        tf3=new JTextField();
        tf3.setBounds(50,150,150,20);
        ex
        b1=new JButton("+");
        b1.setBounds(50,200,50,50);
        b2=new JButton("-");
        b2.setBounds(120,200,50,50);
        b1.addActionListener(this);
        b2.addActionListener(this);
        f.add(tf1);f.add(tf2);f.add(tf3);f.add(b1);f.add(b2);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        String s1=tf1.getText();
        String s2=tf2.getText();
        int a=Integer.parseInt(s1);
        int b=Integer.parseInt(s2);
        int c=0;
        if(e.getSource()==b1){
            c=a+b;
        }else if(e.getSource()==b2){
            c=a-b;
        }
        String result=String.valueOf(c);
        tf3.setText(result);
    }
public static void main(String[] args) {
    new TextFieldExample();
```

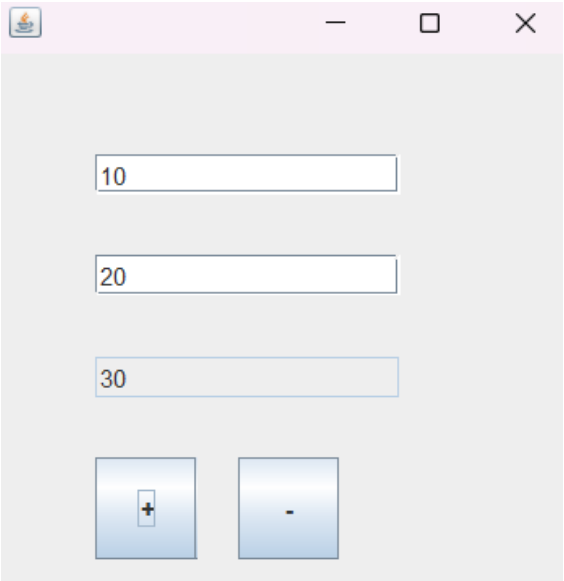
```
}}
```

Output:

At first, the program will give the following output asking for two inputs.



Once we type two numbers such as 10 and 20 and then press on the “+” button, it will show the output as follows.



1.4 JTextArea class

The object of a JTextArea class is a multi line region that displays text. It allows the editing of multiple line text. It inherits JTextComponent class

1.4.1 Commonly used Constructors:

The most commonly used constructors are shown in the table below.

Constructor	Description

JTextArea()	Creates a text area that displays no text initially.
JTextArea(String s)	Creates a text area that displays specified text initially.
JTextArea(int row, int column)	Creates a text area with the specified number of rows and columns that displays no text initially.

Table 7: The most commonly used constructors

1.4.2 Commonly used Methods:

The most commonly used methods are shown in the table below.

Methods	**Description**
void setRows(int rows)	It is used to set specified number of rows.
void setColumns(int cols)	It is used to set specified number of columns.
void setFont(Font f)	It is used to set the specified font.

void insert(String s, int position)	It is used to insert the specified text on the specified position.
void append(String s)	It is used to append the given text to the end of the document.

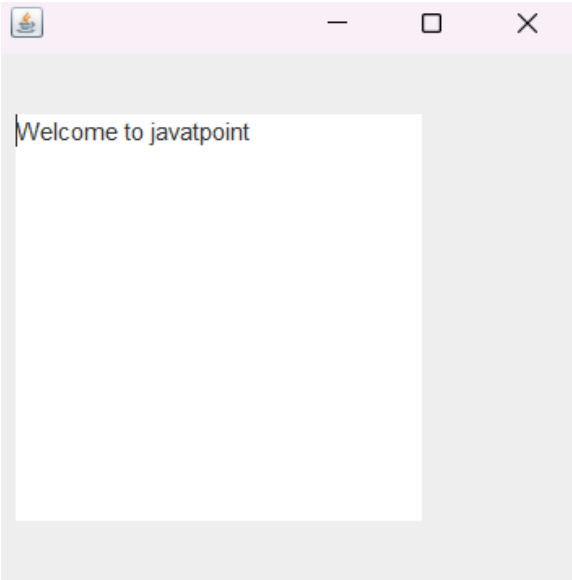
Table 8: The most commonly used methods

1.4.4 Java JTextArea Example

Program1:

```
import javax.swing.*;
public class TextAreaExample
{
    TextAreaExample(){
      JFrame f= new JFrame();
      JTextArea area=new JTextArea("Welcome to javatpoint");
      area.setBounds(10,30, 200,200);
      f.add(area);
      f.setSize(300,300);
      f.setLayout(null);
      f.setVisible(true);
    }
public static void main(String args[])
   {
  new TextAreaExample();
   }}
```

Output:



1.4.5 Java JTextArea Example with ActionListener

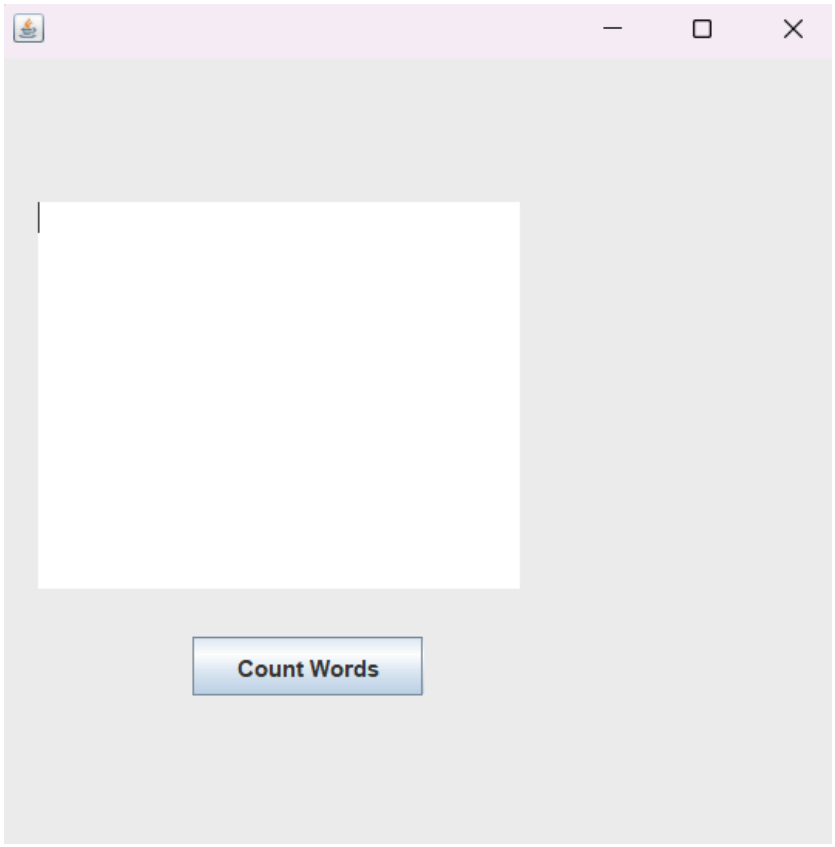
Program1:

```
import javax.swing.*;
import java.awt.event.*;
public class TextAreaExample implements ActionListener{
JLabel l1,l2;
JTextArea area;
JButton b;
TextAreaExample() {
    JFrame f= new JFrame();
    l1=new JLabel();
    l1.setBounds(50,25,100,30);
    l2=new JLabel();
    l2.setBounds(160,25,100,30);
    area=new JTextArea();
    area.setBounds(20,75,250,200);
    b=new JButton("Count Words");
    b.setBounds(100,300,120,30);
```

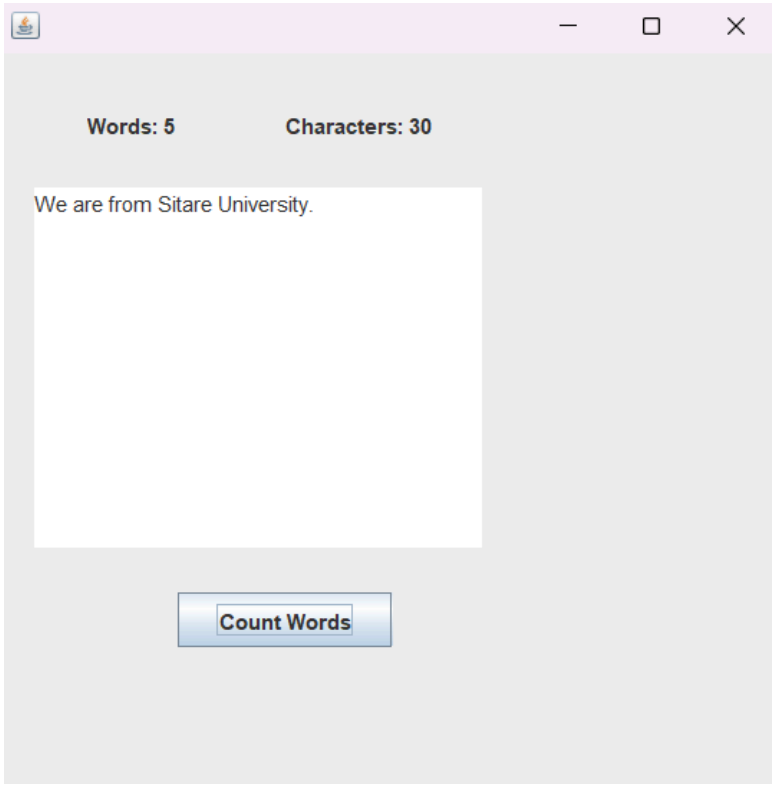
```
    b.addActionListener(this);
    f.add(l1);f.add(l2);f.add(area);f.add(b);
    f.setSize(450,450);
    f.setLayout(null);
    f.setVisible(true);
}
public void actionPerformed(ActionEvent e){
    String text=area.getText();
    String words[]=text.split("\\s");
    l1.setText("Words: "+words.length);
    l2.setText("Characters: "+text.length());
}
public static void main(String[] args) {
    new TextAreaExample();
}
}
```

Output:

Firstly it will show the following output.



Once we type a text and then click on the “count words” button, it will produce the following output.


Words: 5
Characters: 30
We are from Sitare University.
Count Words