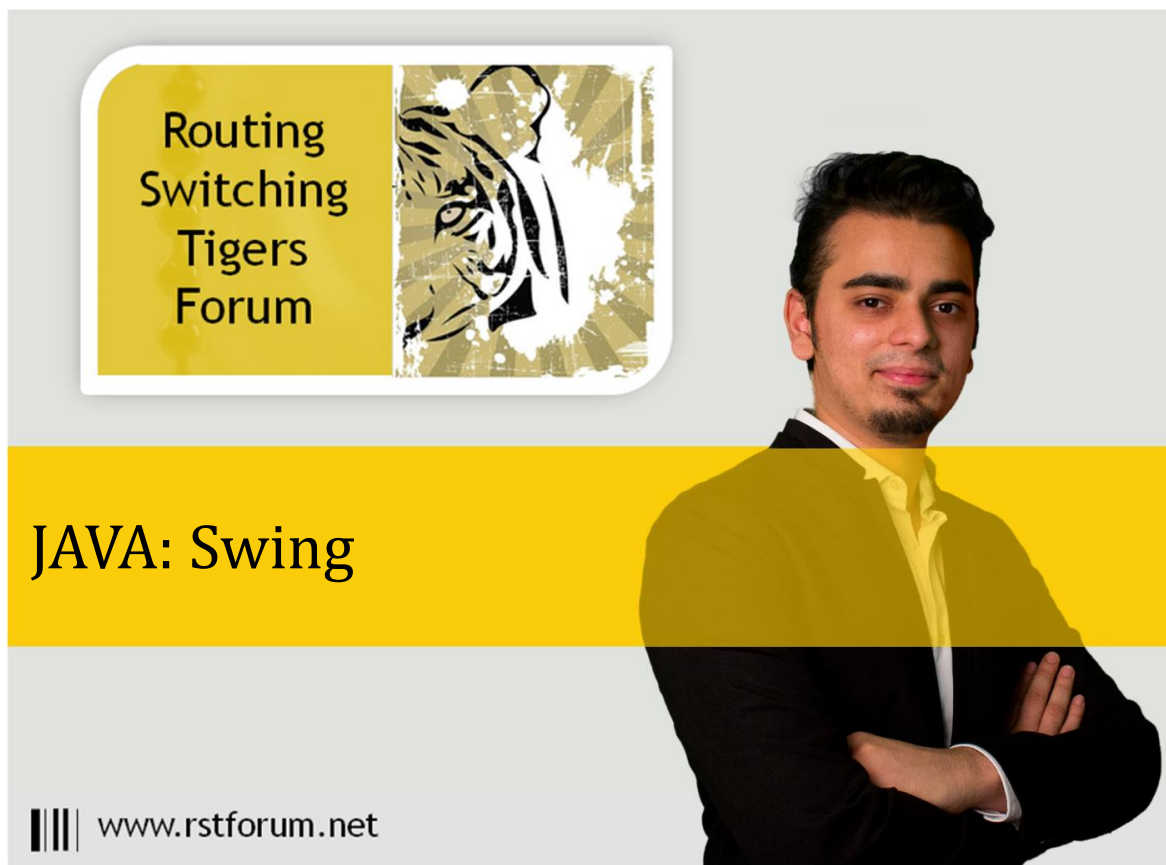


LAB11: JAVA SWING

Disclaimer

This Configuration Guide is designed to assist members to enhance their skills in respective technology area. While every effort has been made to ensure that all material is as complete and accurate as possible, the enclosed material is presented on an “as is” basis. Neither the authors nor Forum assume any liability or responsibility to any person or entity with respect to loss or damages incurred from the information contained in this guide. This Lab Guide was developed by RSTForum. Any similarities between material presented in this configuration guide and any other material is completely coincidental.



Routing
Switching
Tigers
Forum

JAVA: Swing

www.rstforum.net

LAB OBJECTIVE: To gain ability to create graphical user interface (GUI) components and build mini application

JAVA Swing:

Java Swing is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.

Task 1: Let's see a simple swing example where we are creating one button and adding it on the JFrame object inside the main() method.

```
import javax.swing.*;

public class FirstSwingExample {

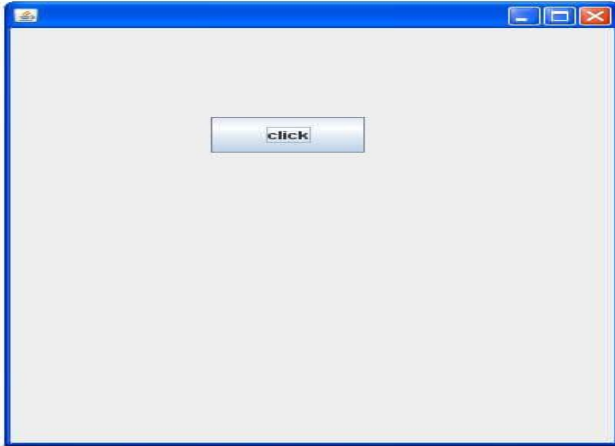
    public static void main(String[] args) {

        JFrame f=new JFrame();           //creating instance of JFrame
        JButton b=new JButton("click"); //creating instance of JButton
        b.setBounds(130,100,100, 40);   // (x axis, y axis, width, height )
        f.add(b);                        //adding button in JFrame
        f.setSize(400,500);              //400 width and 500 height
        f.setLayout(null);               //using no layout managers
        f.setVisible(true);              //making the frame visible
    }
}
```

Output:

```
Javac FirstSwingExample.java
```

```
Java FirstSwingExample
```



Task 2: Creating a Button with ActionListener

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

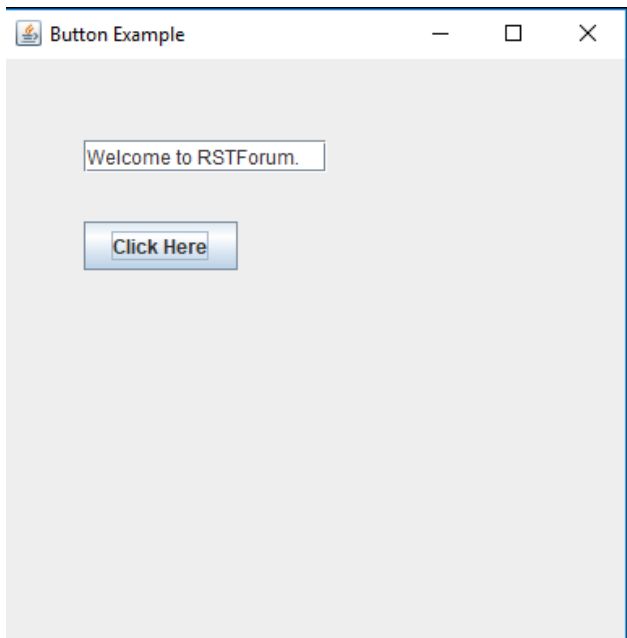
public class ButtonExample1 {
    public static void main(String[] args) {
        JFrame f=new JFrame("Button Example");
        final JTextField tf=new JTextField();
        tf.setBounds(50,50, 150,20);
        JButton b=new JButton("Click Here");
        b.setBounds(50,100,95,30);
        b.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                tf.setText("Welcome to RSTForum.");
            }
        });
        f.add(b);f.add(tf);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
```

```
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // used to close the window  
}  
}
```

Output:

Javac ButtonExample1.java

Java ButtonExample1



Task 3: Example of Swing by Association inside constructor

We can also write all the codes of creating JFrame, JButton and method call inside the java constructor

```
import javax.swing.*;  
public class Simple {  
    JFrame f;  
    Simple(){  
        f=new JFrame(); //creating instance of JFrame
```

```
 JButton b=new JButton("click");//creating instance of JButton  
 b.setBounds(130,100,100, 40);
```

```
 f.add(b);//adding button in JFrame
```

```
 f.setSize(400,500);//400 width and 500 height
```

```
 f.setLayout(null);//using no layout managers
```

```
 f.setVisible(true);//making the frame visible
```

```
 }
```

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

```
 new Simple();
```

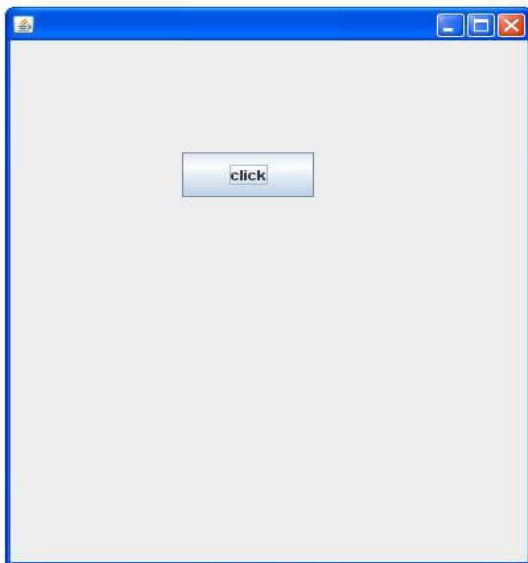
```
 }
```

```
 }
```

Output:

```
 Javac Simple.java
```

```
 Java Simple
```



Task 4: Simple example of Swing by inheritance:

We can also inherit the JFrame class, so there is no need to create the instance of JFrame class explicitly.

```
import javax.swing.*;

public class Simple2 extends JFrame{ //inheriting JFrame
    JFrame f;

    Simple2(){

        JButton b=new JButton("click");//create button

        b.setBounds(130,100,100, 40);

        add(b);//adding button on frame

        setSize(400,500);

        setLayout(null);

        setVisible(true);

    }

    public static void main(String[] args) {

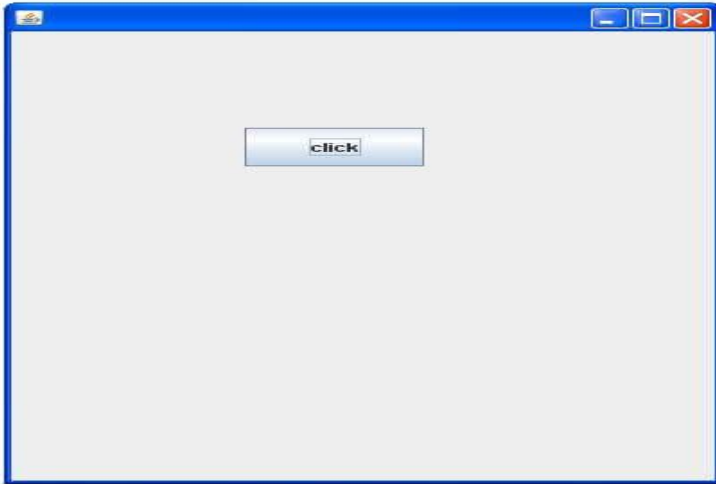
        new Simple2();

    }
}
```

Output:

```
Javac Simple2.java
```

```
Java Simple2
```



Task 5: Create a Button using JButton class

```
import javax.swing.*;

public class ButtonExample {

    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);

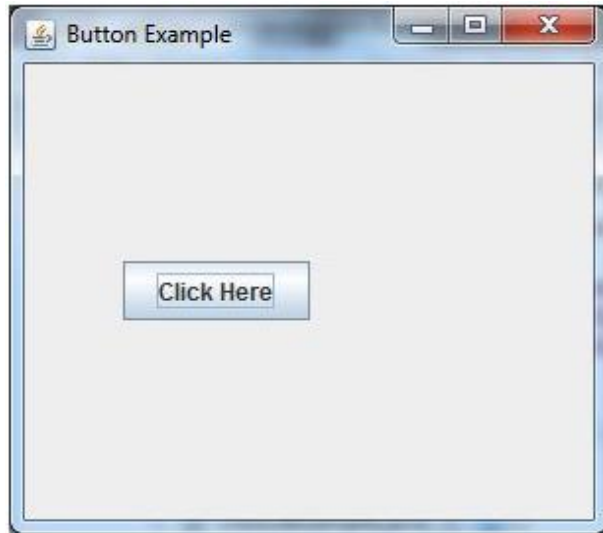
    }

}
```

Output:

Javac ButtonExample.java

Java ButtonExample



Task 6: Create Button with ActionListener:

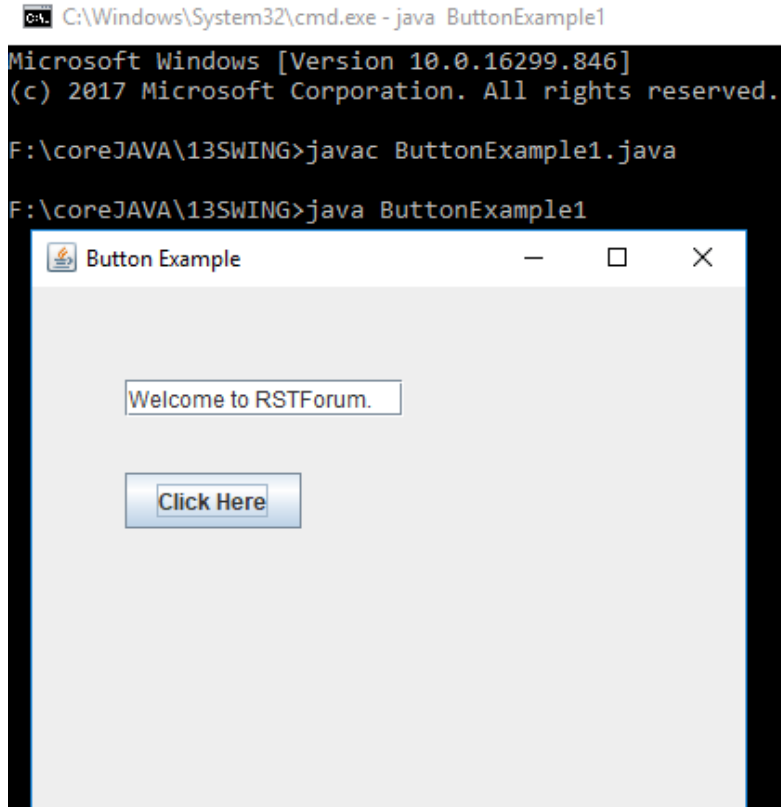
```
import java.awt.event.*;
import javax.swing.*;

public class ButtonExample1 {
    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("Welcome to RSTForum.");
            }
        });
        f.add(b);f.add(tf);
        f.setSize(400,400);
```



```
f.setLayout(null);  
f.setVisible(true);  
}  
}
```

Output:



Task 7: Display the image on the button

```
import javax.swing.*;  
public class ButtonExample2{  
    ButtonExample2(){  
        JFrame f=new JFrame("Button Example");  
        JButton b=new JButton(new ImageIcon("D:\\icon.png")); // add image on button
```

```
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 ButtonExample2();  
}  
}
```

Output:

Javac ButtonExample2.java

Java ButtonExample2



Java JLabel Example:

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.

Task 8: To create labels on Frame

```
import javax.swing.*;

class LabelExample
{
public static void main(String args[])
    {
    JFrame f= new JFrame("Label Example");

    JLabel l1,l2;

    l1=new JLabel("First Label."); // creating instance of 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);

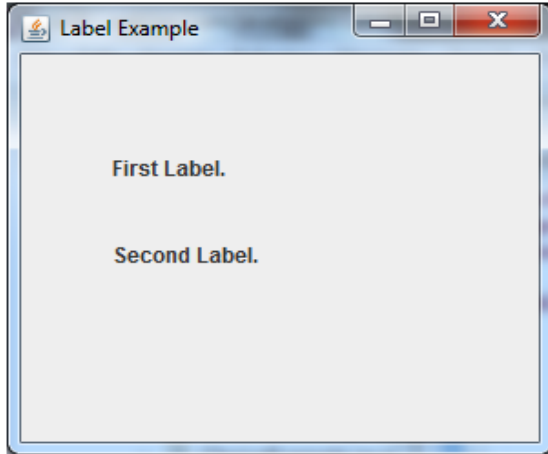
    f.setVisible(true); }

}
```

Output:

```
Javac LabelExample .java
```

```
Java LabelExample
```



Task 9: Java JLabel Example with ActionListener:

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

public class LabelExample1 extends Frame implements ActionListener{

    JTextField tf; JLabel l; JButton b;

    LabelExample1(){

        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 LabelExample1();  
}}
```

Output:



Java JTextField:

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

Task 10: Create TextField Using JTextField class

```
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 RSTForum.");

        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);

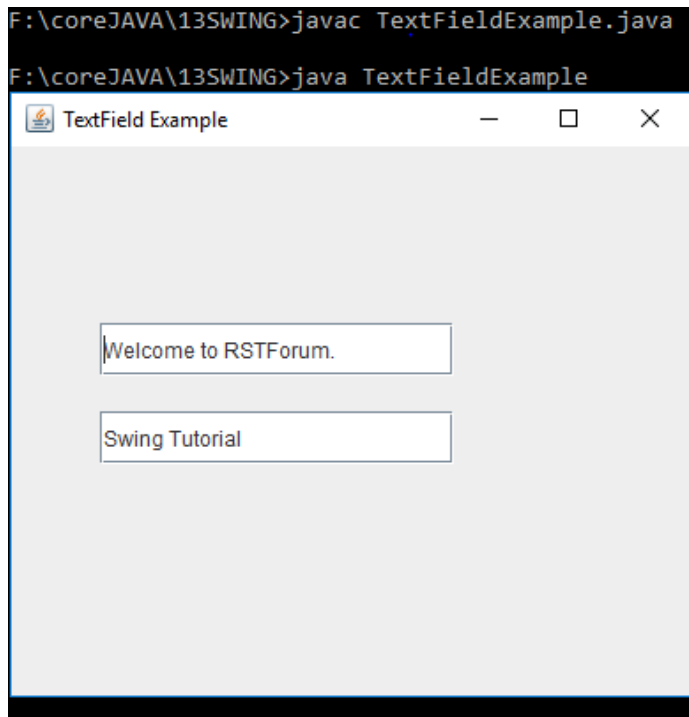
        f.setLayout(null);

        f.setVisible(true);

    }

}
```

Output:



Task 11: Create TextField with ActionListener:

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

public class TextFieldExample1 implements ActionListener{

    JTextField tf1,tf2,tf3;

    JButton b1,b2;

    TextFieldExample1(){

        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);
```

```

tf3.setEditable(false);

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);

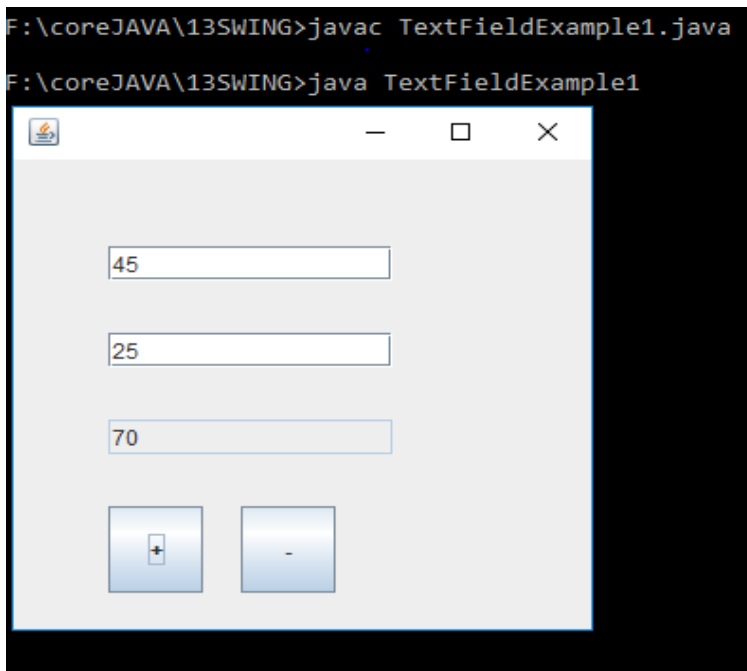
}

```



```
public static void main(String[] args) {  
    new TextFieldExample1();  
}}
```

Output:



Java JTextArea:

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

Task 12: To create Text Area using JTextArea class

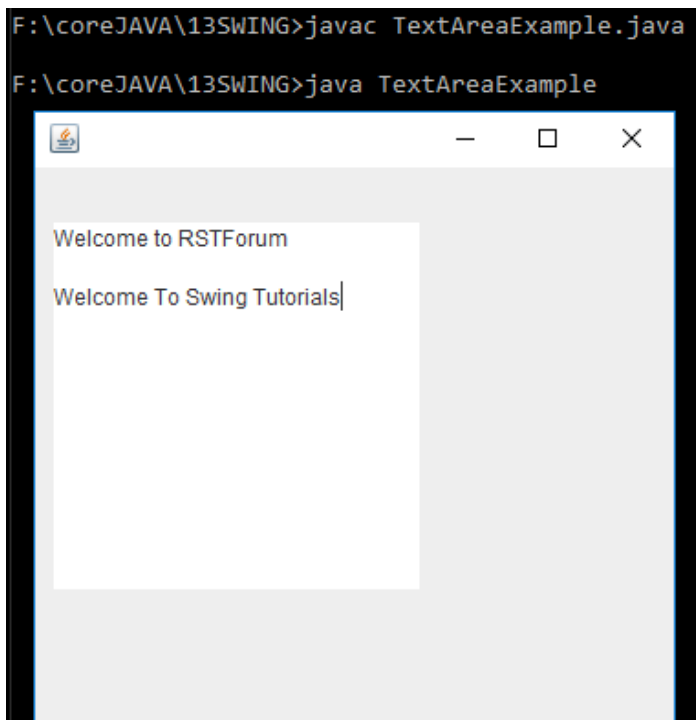
```
import javax.swing.*;  
  
public class TextAreaExample  
{  
    TextAreaExample(){  
        JFrame f= new JFrame();  
        JTextArea area=new JTextArea("Welcome to javatpoint");  
        area.setBounds(10,30, 200,200);
```

```
f.add(area);  
f.setSize(300,300);  
f.setLayout(null);  
f.setVisible(true);  
}  
public static void main(String args[])  
{  
    new TextAreaExample();  
}}
```

Output:

Javac TextAreaExample.java

Java TextAreaExample



Task 13: Create TextArea with ActionListener

```

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

public class TextAreaExample1 implements ActionListener{
    JLabel l1,l2;
    JTextArea area;
    JButton b;

    TextAreaExample1() {
        JFrame f= new JFrame("TextAreaExample1");
        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(400,450);
        f.setLayout(null);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);    // close windows
    }

    public void actionPerformed(ActionEvent e){ // automatically invoked

        String text=area.getText();

        String words[]=text.split("\\s"); // split() method to split string based on space
    }
}

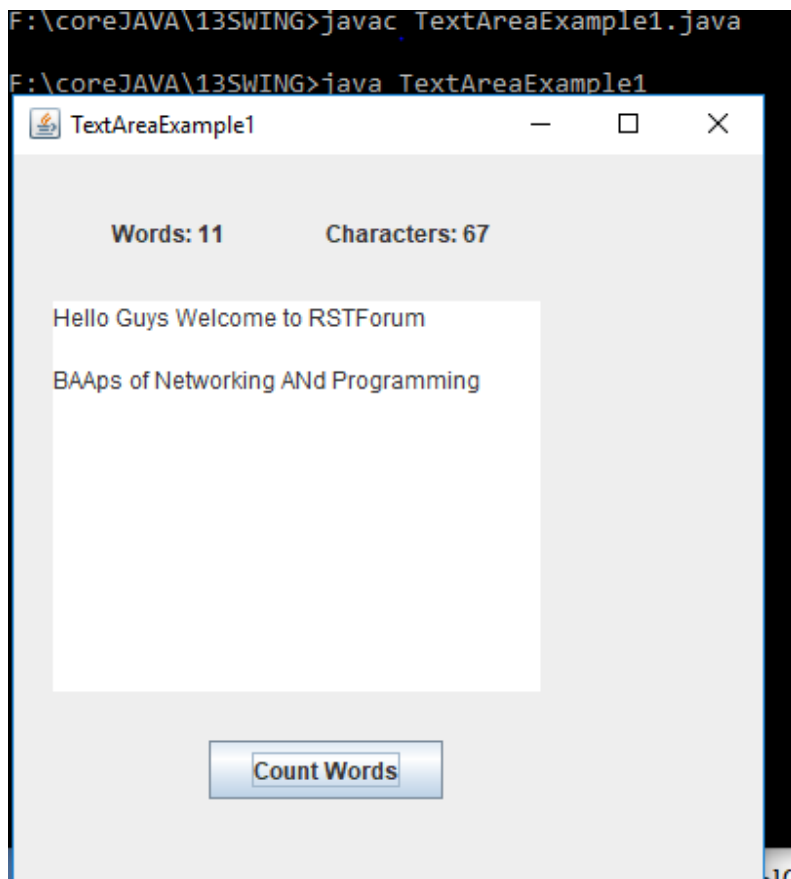
```

```
l1.setText("Words: "+words.length);  
l2.setText("Characters: "+text.length());  
}  
public static void main(String[] args) {  
    new TextAreaExample1();  
}  
}
```

Output:

javac TextAreaExample1.java

java TextAreaExample1



Java JPasswordField:

The object of a JPasswordField class is a text component specialized for password entry. It allows the editing of a single line of text. It inherits JTextField class.

Task 13: Create PasswordField using JPasswordField class:

```
import javax.swing.*;

public class PasswordFieldExample {

    public static void main(String[] args) {

        JFrame f=new JFrame("Password Field Example");

        JPasswordField value = new JPasswordField();

        JLabel l1=new JLabel("Password:");

        l1.setBounds(20,100, 80,30);

        value.setBounds(100,100,100,30);

        f.add(value); f.add(l1);

        f.setSize(300,300);

        f.setLayout(null);

        f.setVisible(true);

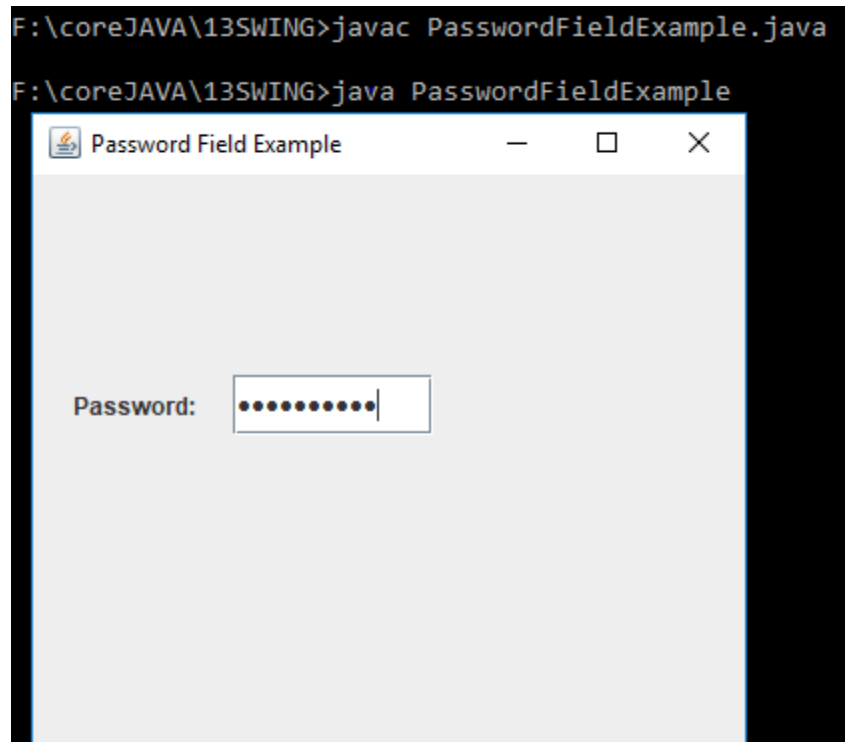
    }

}
```

Output:

```
Javac PasswordFieldExample.java
```

```
Java PasswordFieldExample
```



Task 14: Create PasswordField with ActionListener

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

public class PasswordFieldExample1{
    public static void main(String[] args) {
        JFrame f=new JFrame("Password Field Example");
        final JLabel label = new JLabel();
        label.setBounds(20,150, 200,50);
        final JPasswordField value = new JPasswordField();
        value.setBounds(100,75,100,30);
        JLabel l1=new JLabel("Username:");
        l1.setBounds(20,20, 80,30);
```

```

JLabel l2=new JLabel("Password:");
l2.setBounds(20,75, 80,30);
JButton b = new JButton("Login");
b.setBounds(100,120, 80,30);
final JTextField text = new JTextField();
text.setBounds(100,20, 100,30);

    f.add(value); f.add(l1); f.add(label); f.add(l2); f.add(b); f.add(text);

    f.setSize(300,300);

    f.setLayout(null);

    f.setVisible(true);

    b.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        String data = "Username " + text.getText();

        data += ", Password: "

        + new String(value.getPassword());

        label.setText(data);

    }

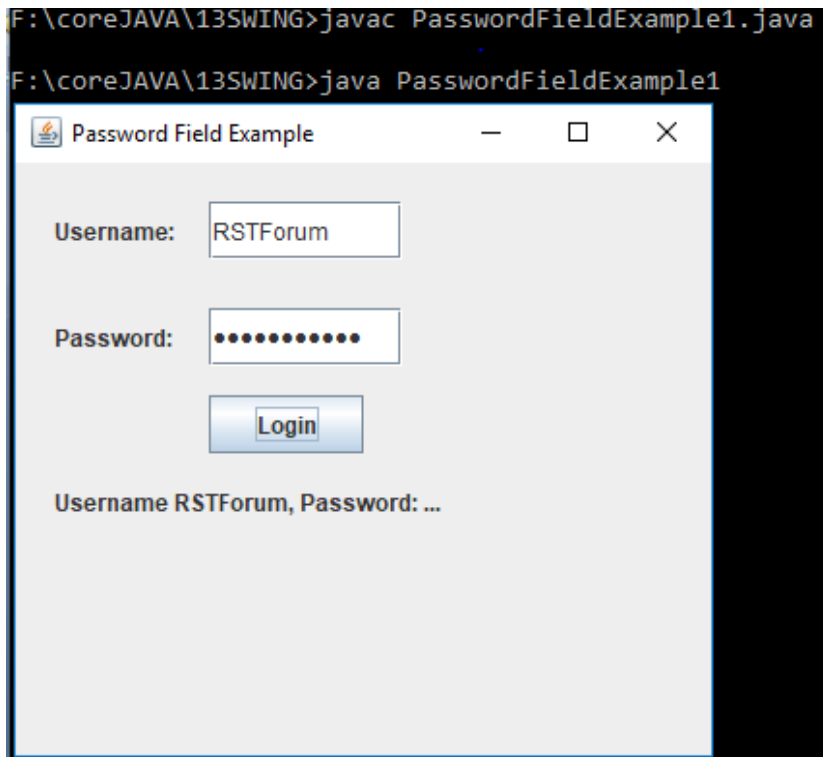
    });

}

}

```

Output:



Java JCheckBox:

The JCheckBox class is used to create a checkbox. It is used to turn an option on (true) or off (false). Clicking on a CheckBox changes its state from "on" to "off" or from "off" to "on ".It inherits JToggleButton class.

Task 15: Create CheckBox on the frame

```
import javax.swing.*;

public class CheckBoxExample
{
    CheckBoxExample(){
        JFrame f= new JFrame("CheckBox Example");
        JCheckBox checkBox1 = new JCheckBox("C++");
        checkBox1.setBounds(100,100, 50,50);
        JCheckBox checkBox2 = new JCheckBox("Java", true);
```



```
checkbox2.setBounds(100,150, 50,50);

f.add(checkbox1);

f.add(checkbox2);

f.setSize(400,400);

f.setLayout(null);

f.setVisible(true);

}

public static void main(String args[])

{

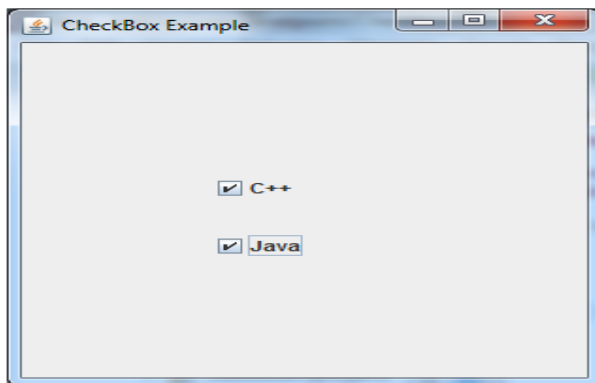
new CheckBoxExample();

}}
```

Output:

Javac CheckBoxExample.java

Java CheckBoxExample



Task 16: Create checkbox with message

Java JCheckBox Example with ItemListener

```
import javax.swing.*;

import java.awt.event.*;

public class CheckBoxExample
```

```

{
    CheckBoxExample1(){
        JFrame f= new JFrame("CheckBox Example");
        final JLabel label = new JLabel();
        label.setHorizontalAlignment(JLabel.CENTER);
        label.setSize(400,100);
        JCheckBox checkbox1 = new JCheckBox("C++");
        checkbox1.setBounds(150,100, 50,50);
        JCheckBox checkbox2 = new JCheckBox("Java");
        checkbox2.setBounds(150,150, 50,50);
        f.add(checkbox1); f.add(checkbox2); f.add(label);
        checkbox1.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                label.setText("C++ Checkbox: "
                    + (e.getStateChange()==1?"checked":"unchecked"));
            }
        });
        checkbox2.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                label.setText("Java Checkbox: "
                    + (e.getStateChange()==1?"checked":"unchecked"));
            }
        });
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
}

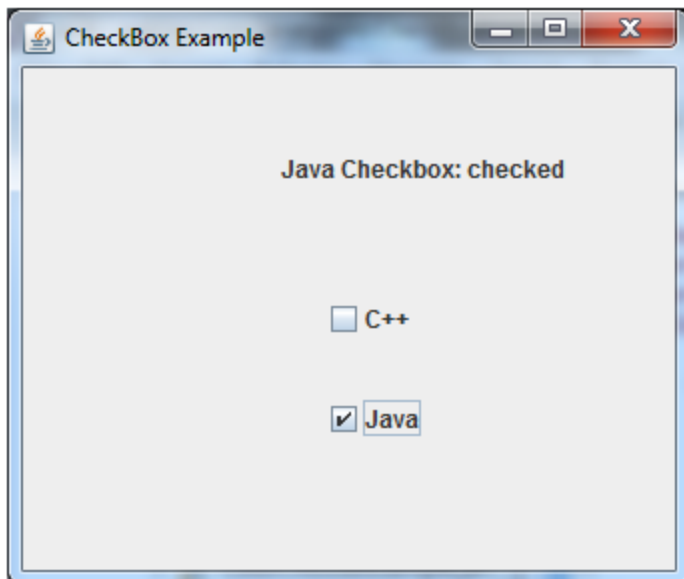
```

```
    }  
    public static void main(String args[])  
    {  
        new CheckBoxExample1();  
    }  
}
```

Output:

Javac CheckBoxExample1.java

Java CheckBoxExample1



Task 17: Create a mini application to Food Delivery using CheckBox

```
import javax.swing.*;  
import java.awt.event.*;  
public class CheckBoxExample extends JFrame implements ActionListener{  
    JLabel l;
```

```

JCheckBox cb1,cb2,cb3;

JButton b;

CheckBoxExample(){

    l=new JLabel("Food Ordering System");

    l.setBounds(50,50,300,20);

    cb1=new JCheckBox("Pizza @ 100");

    cb1.setBounds(100,100,150,20);

    cb2=new JCheckBox("Burger @ 30");

    cb2.setBounds(100,150,150,20);

    cb3=new JCheckBox("Tea @ 10");

    cb3.setBounds(100,200,150,20);

    b=new JButton("Order");

    b.setBounds(100,250,80,30);

    b.addActionListener(this);

    add(l);add(cb1);add(cb2);add(cb3);add(b);

    setSize(400,400);

    setLayout(null);

    setVisible(true);

    setDefaultCloseOperation(EXIT_ON_CLOSE);

}

public void actionPerformed(ActionEvent e){

    float amount=0;

    String msg="";

    if(cb1.isSelected()){

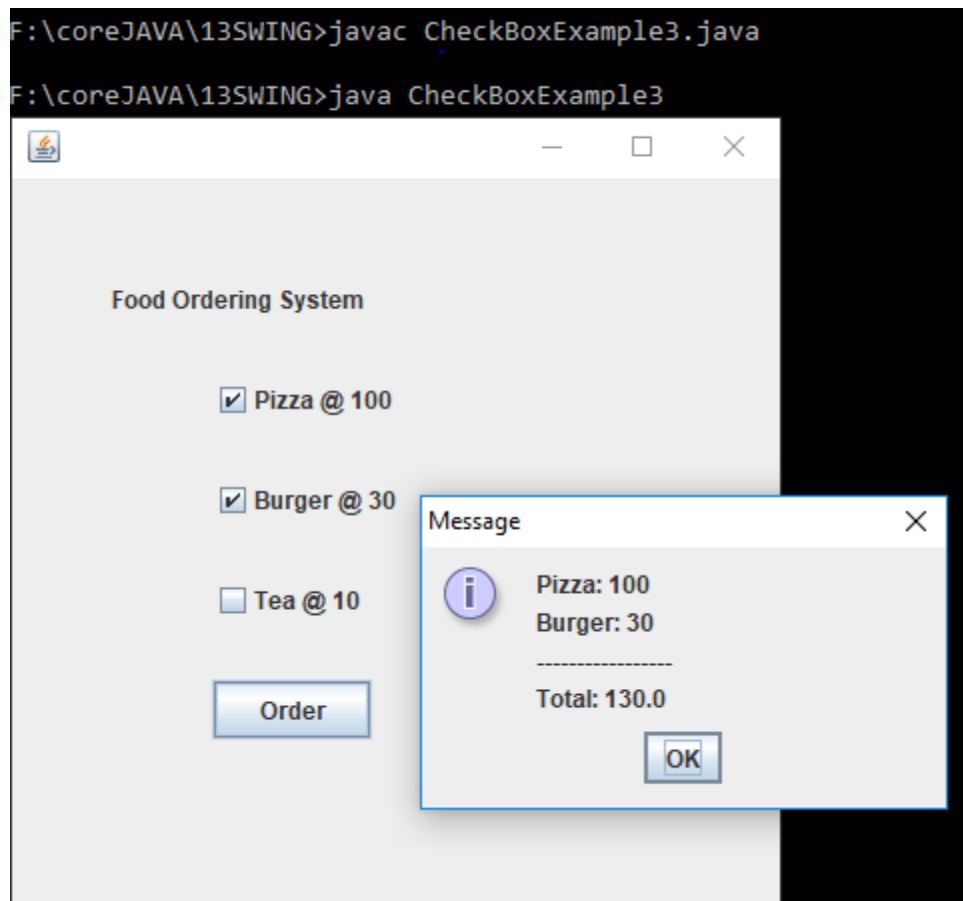
        amount+=100;

        msg="Pizza: 100\n";
    }
}

```

```
}
if(cb2.isSelected()){
    amount+=30;
    msg+="Burger: 30\n";
}
if(cb3.isSelected()){
    amount+=10;
    msg+="Tea: 10\n";
}
msg+="-----\n";
OptionPane.showMessageDialog(this,msg+"Total: "+amount);
}
public static void main(String[] args) {
    new CheckBoxExample();
}
}
```

Output:



Java JComboBox

The object of Choice class is used to show popup menu of choices. Choice selected by user is shown on the top of a menu. It inherits JComponent class.

Task 18: Create JComboBox with ActionListener

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

public class ComboBoxExample1 {
    JFrame f;

    ComboBoxExample1(){
        f=new JFrame("ComboBox Example");
```

```

final JLabel label = new JLabel();

label.setHorizontalAlignment(JLabel.CENTER);

label.setSize(400,100);

JButton b=new JButton("Show");

b.setBounds(200,100,75,20);

String languages[]={ "C","C++","C#","Java","PHP"};

final JComboBox cb=new JComboBox(languages);

cb.setBounds(50, 100,90,20);

f.add(cb); f.add(label); f.add(b);

f.setLayout(null);

f.setSize(350,350);

f.setVisible(true);

b.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

String data = "Programming language Selected: "

+ cb.getItemAt(cb.getSelectedIndex());

label.setText(data);

}

});

}

public static void main(String[] args) {

    new ComboBoxExample1();    }

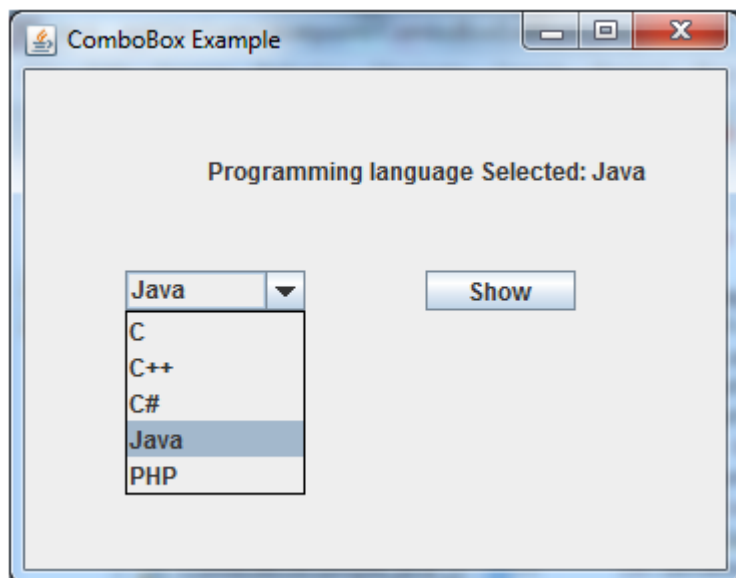
}

```

Output:

Javac ComboBoxExample1.java

Java ComboBoxExample1



The setIconImage() method :

setIconImage() method of Frame class is used to change the icon of Frame or Window. It changes the icon which is displayed at the left side of Frame or Window.

The Toolkit class is used to get instance of Image class in AWT and Swing.

Task 19: To change TitleBar icon in Java AWT:

```
import java.awt.*;

class IconExample {

IconExample(){

Frame f=new Frame();

Image icon = Toolkit.getDefaultToolkit().getImage("E:\\icon2.png");

f.setIconImage(icon);

f.setLayout(null);

f.setSize(400,400);

f.setVisible(true);
```

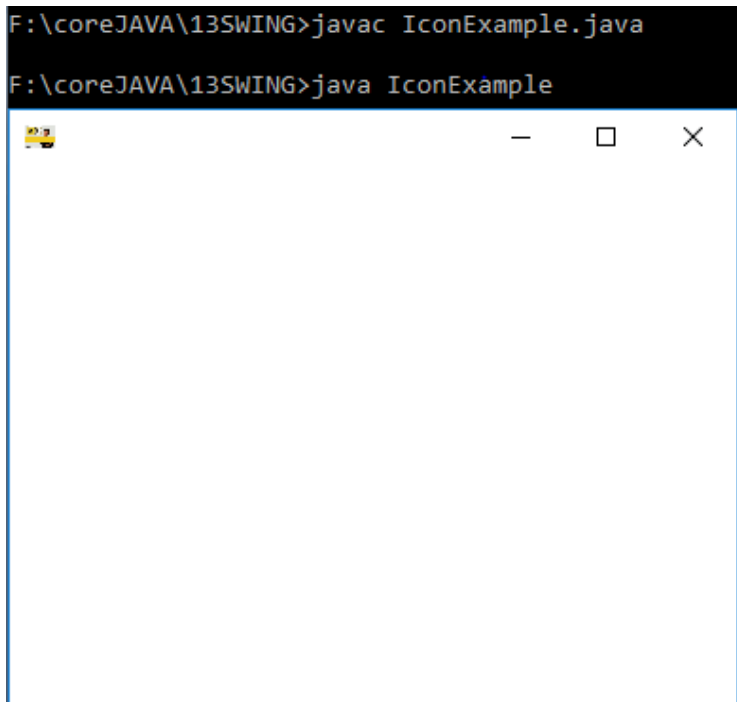


```
}  
public static void main(String args[]){  
new IconExample();  
}  
}
```

Output:

Javac IconExample.java

Java IconExample



Task 20: To change TitleBar icon in Java Swing:

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
class IconExample1{
IconExample1(){
JFrame f=new JFrame();
Image icon = Toolkit.getDefaultToolkit().getImage("E:\\icon2.png");
f.setIconImage(icon);
f.setLayout(null);
f.setSize(400,400);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public static void main(String args[]){
new IconExample1();
}
}
```

Output:

Javac IconExample1.java

Java IconExample1

```
F:\coreJAVA\13SWING>javac IconExample1.java
F:\coreJAVA\13SWING>java IconExample1
```

