Idris Ratlamwala
C31 - 2003145

Experiment 14: Program to create a GUI based Application

Theory :

Swing in java is part of Java foundation class which is lightweight and platform independent. It is used for creating window based applications. It includes components like button, scroll bar, text field etc. Putting together all these components makes a graphical user interface. In this article, we will go through the concepts involved in the process of building applications using swing in Java.

Swing in Java is a lightweight GUI toolkit which has a wide variety of widgets for building optimized window based applications. It is a part of the JFC(Java Foundation Classes). It is build on top of the AWT API and entirely written in java. It is platform independent unlike AWT and has lightweight components.
It becomes easier to build applications since we already have GUI components like button, checkbox etc. This is helpful because we do not have to start from the scratch.

Container Class
Any **class** which has other components in it is called as a container class. For building GUI applications at least one container class is necessary.

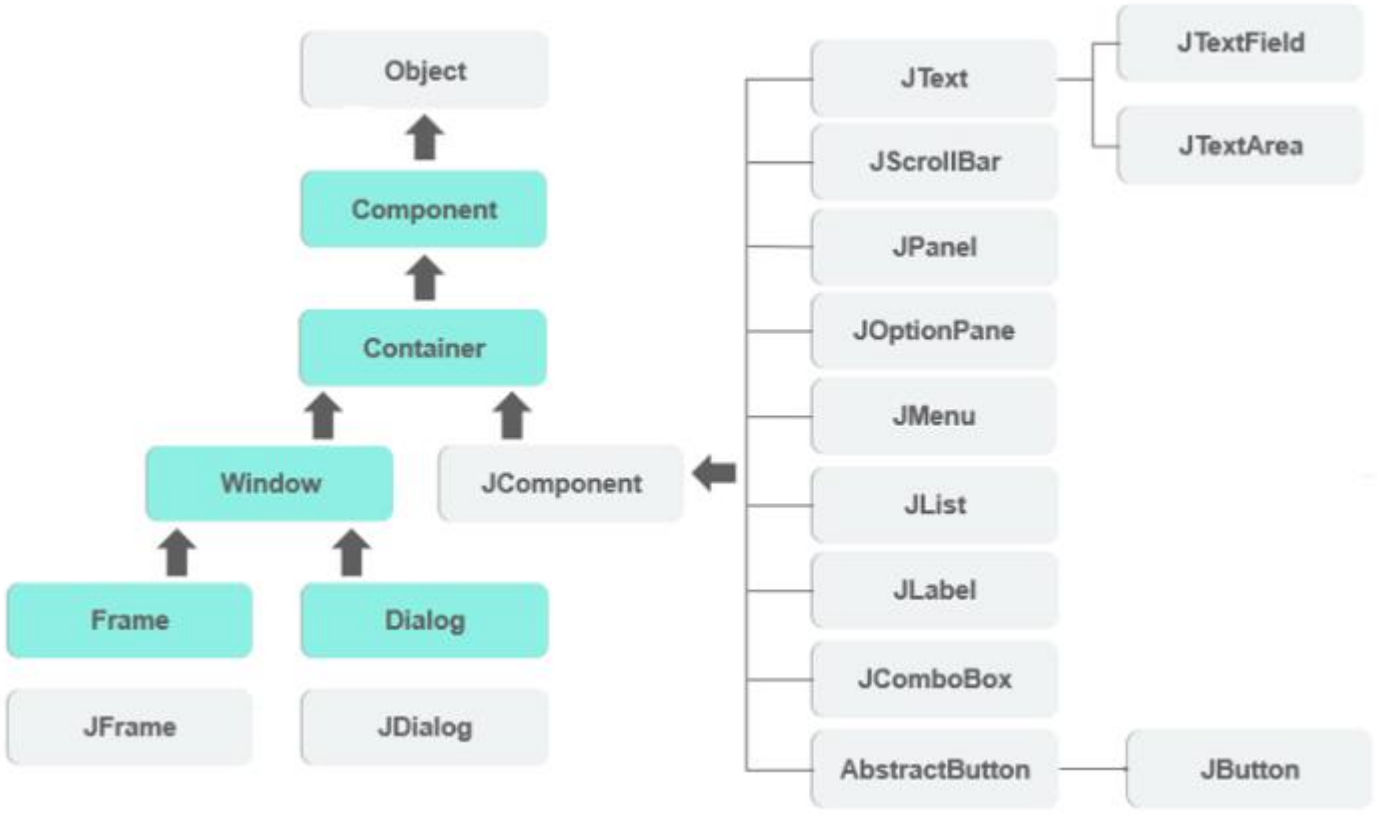
Following are the three types of container classes:

1. Panel – It is used to organize components on to a window
2. Frame – A fully functioning window with icons and titles
3. Dialog – It is like a pop up window but not fully functional like the frame

Difference Between AWT and Swing

AWT	SWING
• Platform Dependent	• Platform Independent
• Does not follow MVC	• Follows MVC
• Lesser Components	• More powerful components
• Does not support pluggable look and feel	• Supports pluggable look and feel
• Heavyweight	• Lightweight

Java Swing Class Hierarchy



All the components in swing like JButton, JComboBox, JList, JLabel are inherited from the JComponent class which can be added to the container classes. Containers are the windows like frame and dialog boxes. Basic swing components are the building blocks of any gui application. Methods like setLayout override the default layout in each container. Containers like JFrame and JDialog can only add a

component to itself. Following are a few components with examples to understand how we can use them.

JButton Class
It is used to create a labelled button. Using the ActionListener it will result in some action when the button is pushed. It inherits the AbstractButton class and is platform independent.

JTextField Class
It inherits the JTextComponent class and it is used to allow editing of single line text.

JPanel Class
It inherits the JComponent class and provides space for an application which can attach any other component.

Java ActionListener Interface

The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. The ActionListener interface is found in java.awt.event package. It has only one method: actionPerformed().
actionPerformed() method

The actionPerformed() method
is invoked automatically whenever you click on the registered component.

```
public abstract void actionPerformed(ActionEvent e);
```

The common approach is to implement the ActionListener. We need to follow 3 steps:

1) Implement the ActionListener interface in the class:

```
public class ActionListenerExample Implements ActionListener
```

2) Register the component with the Listener:

```
component.addActionListener(instanceOfListenerclass);
```

3) Override the actionPerformed() method:

```
1. public void actionPerformed(ActionEvent e){
2.         //Write the code here
3. }
```

A.

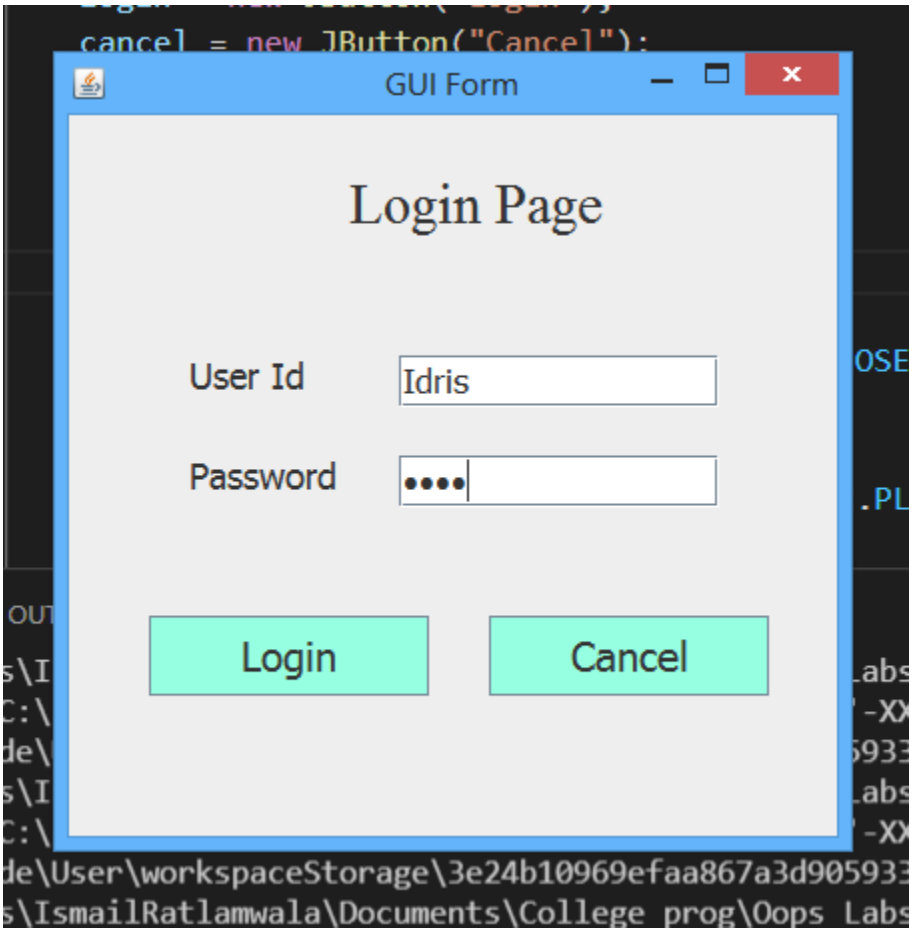
Aim : Create a registration form containing required fields .
The form should have all the studied components
The form should have minimum two buttons “Submit” and “Cancel”….giving appropriate messages at corresponding click.

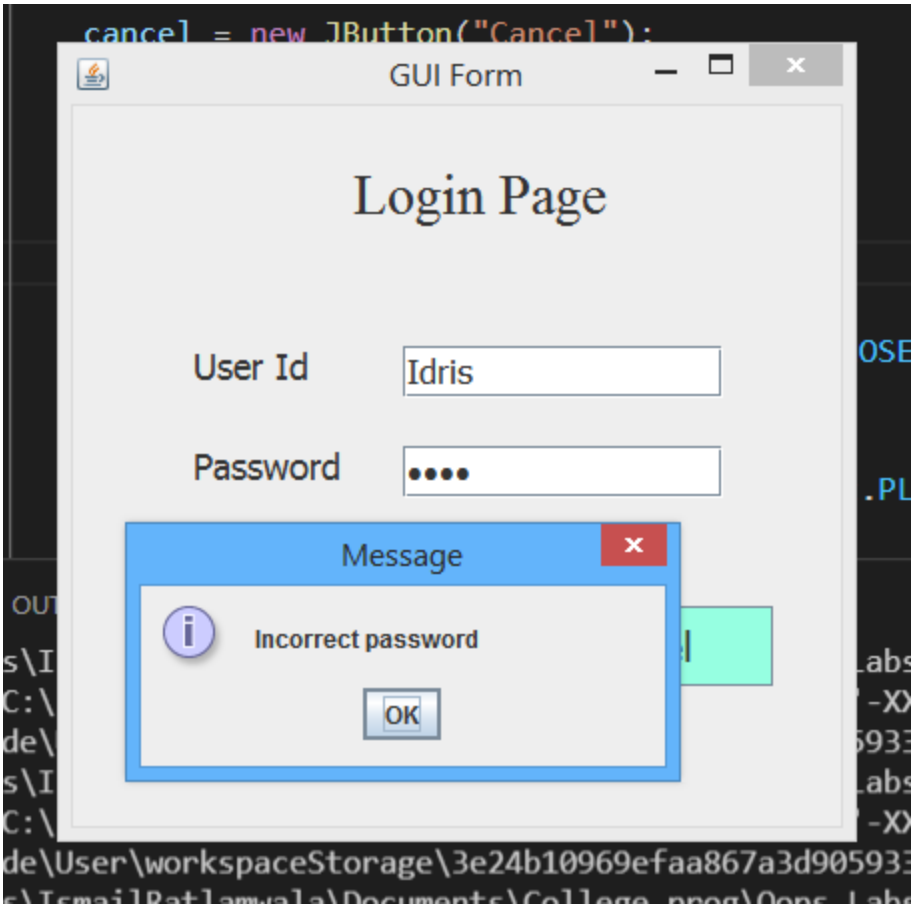
Program :

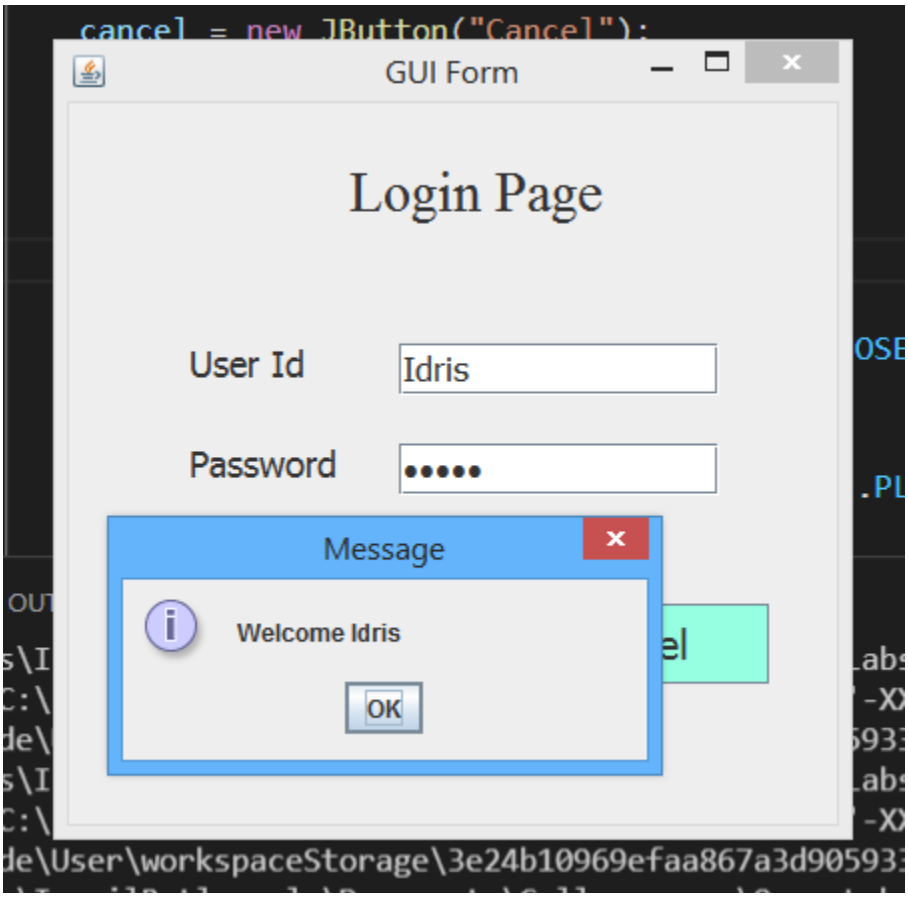
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Login_page {
    public static void main(String[] args) {
        new Windows();
    }
}

class Windows extends JFrame implements ActionListener {
    JTextField password, id;
    JLabel l, l1, l2;
    JButton login,cancel;

    public Windows() {
        password = new JPasswordField(15);
        id = new JTextField(15);
        login = new JButton("Login");
        cancel = new JButton("Cancel");
        l = new JLabel("Login Page");
        l1 = new JLabel("User Id");
        l2 = new JLabel("Password");

        setLayout(null);
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("GUI Form");
        setBounds(200,200,400, 400);
        l.setFont(new Font("Times New Roman", Font.PLAIN, 29));
        l.setBounds(140, 30, 190, 30);

        l1.setFont(new Font("Tahoma", Font.PLAIN, 18));
```

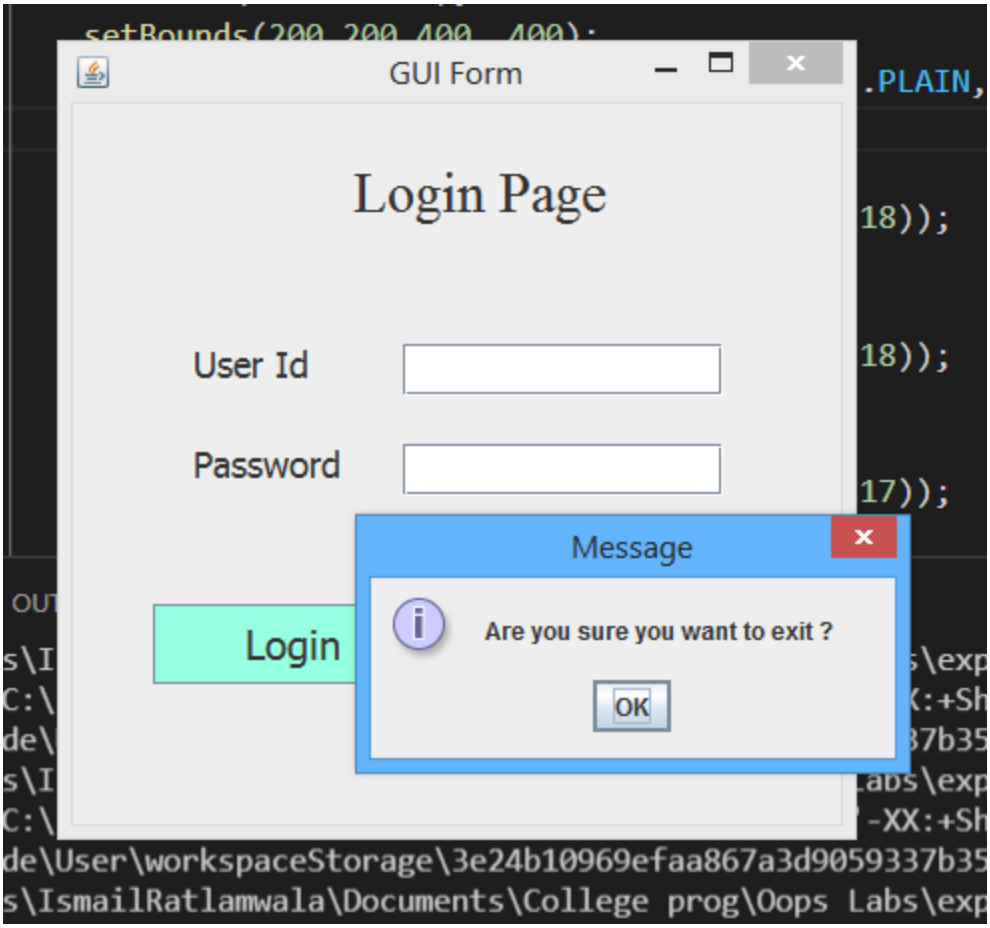
```
        l1.setBounds(60, 120, 80, 20);

        l2.setFont(new Font("Tahoma", Font.PLAIN, 18));
        l2.setBounds(60, 170, 80, 20);

        id.setFont(new Font("Tahoma", Font.PLAIN, 17));
        id.setBounds(165, 120, 160, 26);

        password.setFont(new Font("Tahoma", Font.PLAIN, 17));
        password.setBounds(165, 170, 160, 26);

        login.setFont(new Font("Tahoma", Font.PLAIN, 20));
        login.setBackground(new Color(150, 255, 225));
        login.setBounds(40, 250, 140, 40);
        login.addActionListener(this);

        cancel.setFont(new Font("Tahoma", Font.PLAIN, 20));
        cancel.setBackground(new Color(150, 255, 225));
        cancel.setBounds(210, 250, 140, 40);
        cancel.addActionListener(this);

        add(l);
        add(l1);
        add(id);
        add(l2);
        add(password);
        add(login);
        add(cancel);
    }

    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==login){
            String pass = password.getText();
            String name = id.getText();
            if (pass.equals("hello")) {
                JOptionPane.showMessageDialog(login, "Welcome "+name);
                dispose();
            } else {
                JOptionPane.showMessageDialog(login, "Incorrect password");
            }
        }
        else if(e.getSource()==cancel){
            JOptionPane.showMessageDialog(cancel, "Are you sure you want to exit ?");
            dispose();
        }
    }
}
```

Output :




cancel = new JButton("Cancel");
GUI Form
Login Page
User Id
Idris
Password
Message
Incorrect password
OK
de\User\workspaceStorage\3e24b10969efaa867a3d905933


cancel = new JButton("Cancel");
GUI Form
Login Page
User Id
Idris
Password
Message
Welcome Idris
OK
de\User\workspaceStorage\3e24b10969efaa867a3d90593


GUI Form
Login Page
User Id
Password
Login
Message
Are you sure you want to exit ?
OK