GUI'S

GRAPHICAL USER INTERFACE IN JAVA USING SWING AWT AND WINDOWS BUILDER.

Presented by:
Maleeha afzal
Registration no.
00344
Bese3(B)

There are two sets of Java APIs/ set of libraries for graphics programming:

AWT
API was introduced in JDK 1.0. Most of the AWT components have become obsolete, AWT is very basic.

SWING:
Swing API, a much more comprehensive set of graphics libraries that enhances the AWT.

AWT Packages:

The **java.awt** package contains the core AWT graphics classes:

- GUI Component classes (such as Button, TextField, and Label),
- GUI Container classes (such as Frame, Panel, Dialog and ScrollPane),
- Layout managers (such as FlowLayout, BorderLayout and GridLayout),
- Custom graphics classes (such as Graphics, Color and Font).

The **java.awt.event** package supports event handling:

- Event classes (such as ActionEvent, MouseEvent, KeyEvent and WindowEvent),
- Event Listener Interfaces (such as ActionListener, MouseListener, KeyListener and WindowListener),
- Event Listener Adapter classes (such as MouseAdapter, KeyAdapter, and WindowAdapter).

Importing libraries.

```
import javax.swing.JFrame;
import javax.swing.JPasswordField;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JPanel;
import javax.swing.JButton;
import javax.swing.JCheckBox;
```

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

```
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionListener;//
import java.awt.event.ActionEvent;
```

COMPONENTS:

Frames: (import javax.swing.JFrame)

A frame can also be called as a window. It contains all the other components. There can be a lot of methods to initialize a frame.

The constructor of a class that **extends Jframe class** serves as a frame itself. The components are then added to that frame.

```
import javax.swing.JFrame;
public class Frameclass extends JFrame{
    Frameclass()
    {

    }//constructor
}
```

Frames can also be initialized as follows.

```
JFrame frame1 = new JFrame("FrameDemo");// creating object of JFrame class
frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame1.getContentPane().add(item); //adding other components to frame1.
frame1.setVisible(true);
```

panels: (import javax.swing.JPanel)

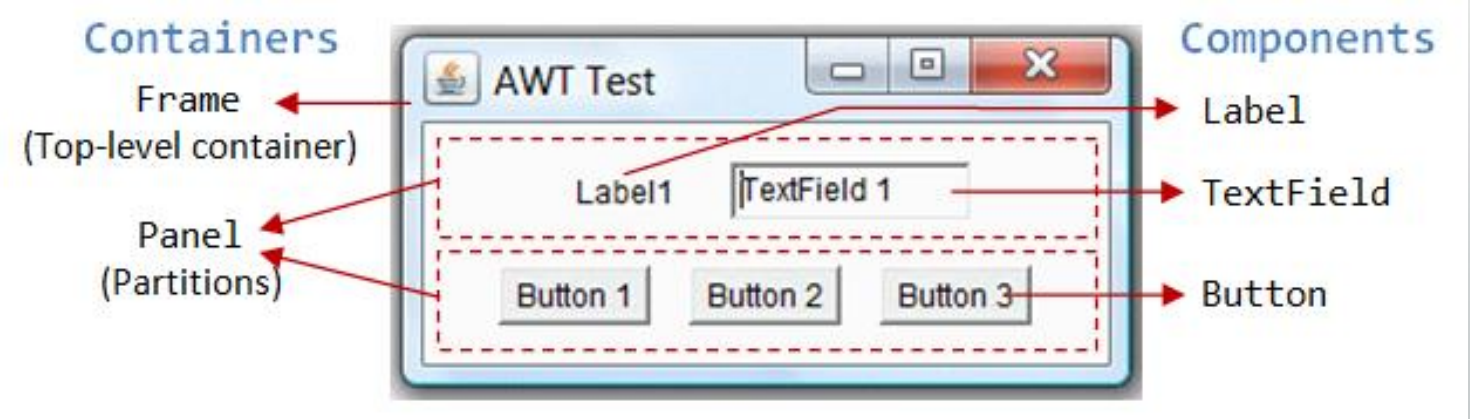
Panels are partitions within a frame that contain other components.

Each panel, like all other components, is an instance of **JPanel** class.

```
JPanel p = new JPanel();
frame1.add(p);
```

Other components are then added to the panel.

```
p.add(item);
```



Text fields: (**import javax.swing.JTextField**)

Can be used for both inputting and outputting plain text.

```
JTextField username = new JTextField(25);
String un=username.getText();
String un1;
username.setText(un1);
```

Password fields: (**import javax.swing.JPasswordField**)

Can be used for bot inputting and outputting passwords.

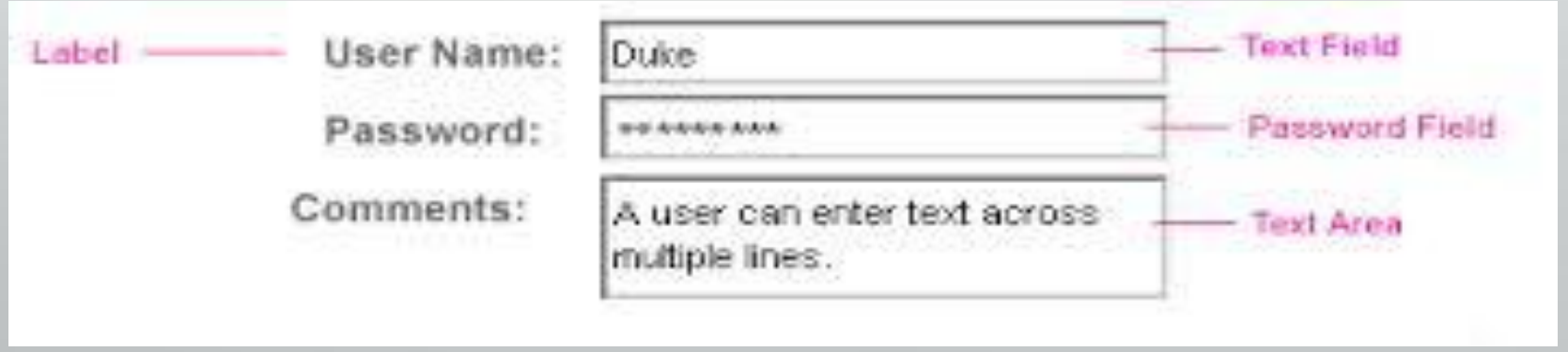
```
JPasswordField pass = new JPasswordField(25);
String s= pass.getText();
String s1;
pass.setText(s1);
```

Text area:

(**import javax.swing.JTextArea**)

For multiple line texts..

```
JTextArea textArea = new JTextArea();
String s2= textArea.getText();
textArea.setText("hello");
```



Buttons: (import javax.swing.JButton)

```
JButton button_8 = new JButton("0");
button_8.setBounds(163, 137, 108, 46);
button_8.setBackground(SystemColor.inactiveCaptionBorder);
panel_1.add(button_8);
```

Combo box: (import javax.swing.JComboBox)

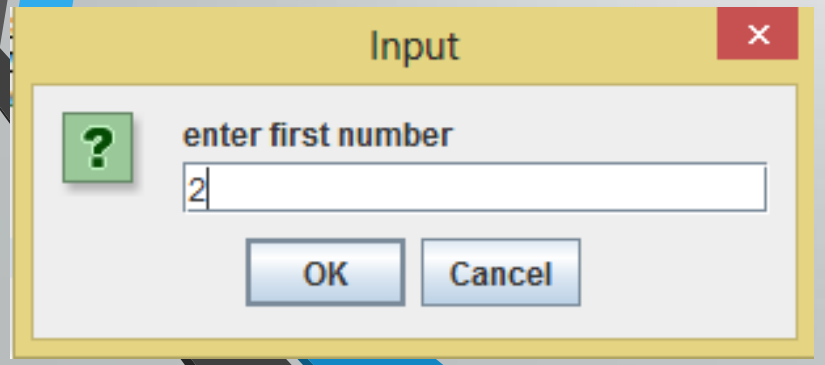
```
String[] petStrings = { "Bird", "Cat", "Dog", "Rabbit", "Pig" };
JComboBox comboBox = new JComboBox(petStrings);
int a= comboBox.getSelectedIndex();
```

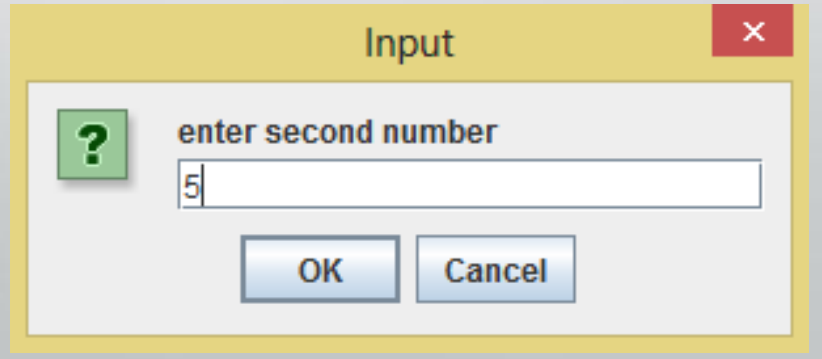
Check box: (import javax.swing.JCheckBox)
tables: (import javax.swing.JTables)
lists: (import javax.swing.JList)

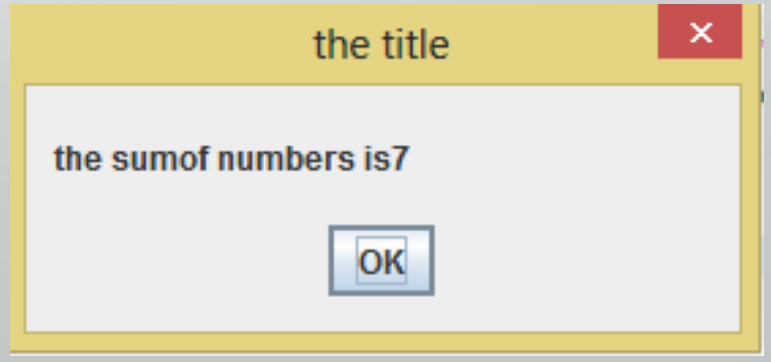
JOptionPane class: (import javax.swing.JOptionPane;)

A class whose methods can be used to initialize other frames built in by default.

```
import javax.swing.JOptionPane; //swing is a gui toolkit.joption pane is the class whose methods we will use.
public class Frameclass {

    public static void main(String[] args)
    {
        String fn=JOptionPane.showInputDialog("enter first number");// showinputdialog and howmessagedialog are met
        String sn=JOptionPane.showInputDialog("enter second number");// displays and input meaasge box and takes in

        int num1= Integer.parseInt(fn);
        int num2= Integer.parseInt(sn);// for conversion from string to int.

        int sum= num1+num2;
        JOptionPane.showMessageDialog(null, "the sumof numbers is" +sum, "the title" , JOptionPane.PLAIN_MESSAGE);
    //argument1 for centering. Arg2 is what we are displaing.Arg3 is title of window. +sum is for accessing variabl
    }

}
```



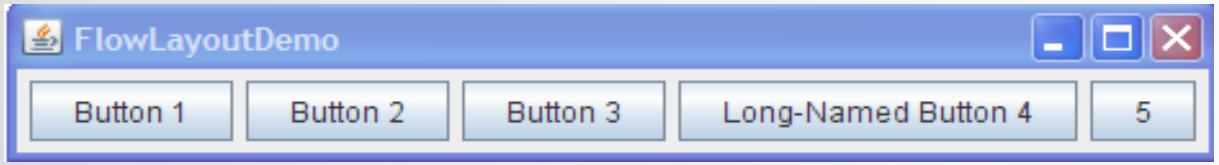




Layouts (for aligning components in a frame or panel).

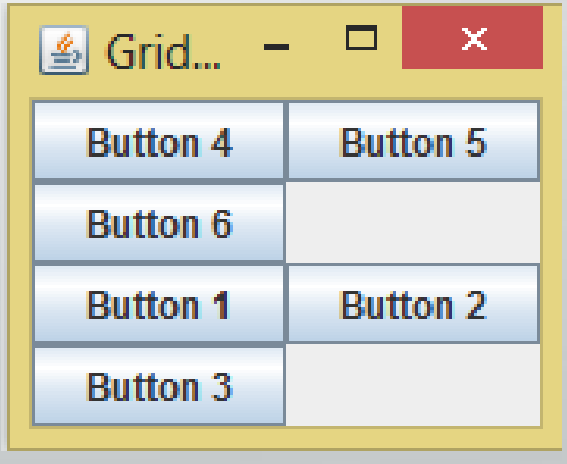
FlowLayout:

```
JPanel loginpanel = new JPanel(new FlowLayout());
```



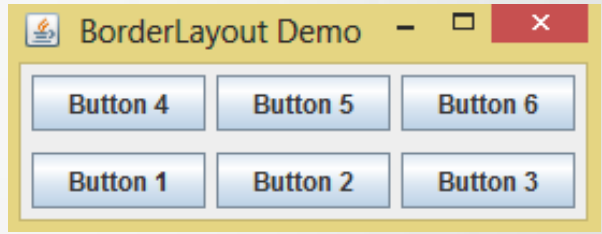
GridLayout:

```
// Creating panel1 and setting it to gridlayout with rows and columns as 2, 2
panel1 = new JPanel( new GridLayout( ROWS, COLUMNS ) );
panel1.add( new JButton( "Button 1" ) );
panel1.add( new JButton( "Button 2" ) );
panel1.add( new JButton( "Button 3" ) );

// creating panel2 and setting to gridlayout with rows and columns as 2, 2
panel2 = new JPanel( new GridLayout( ROWS, COLUMNS ) );
panel2.add( new JButton( "Button 4" ) );
panel2.add( new JButton( "Button 5" ) );
panel2.add( new JButton( "Button 6" ) );
```



BorderLayout:

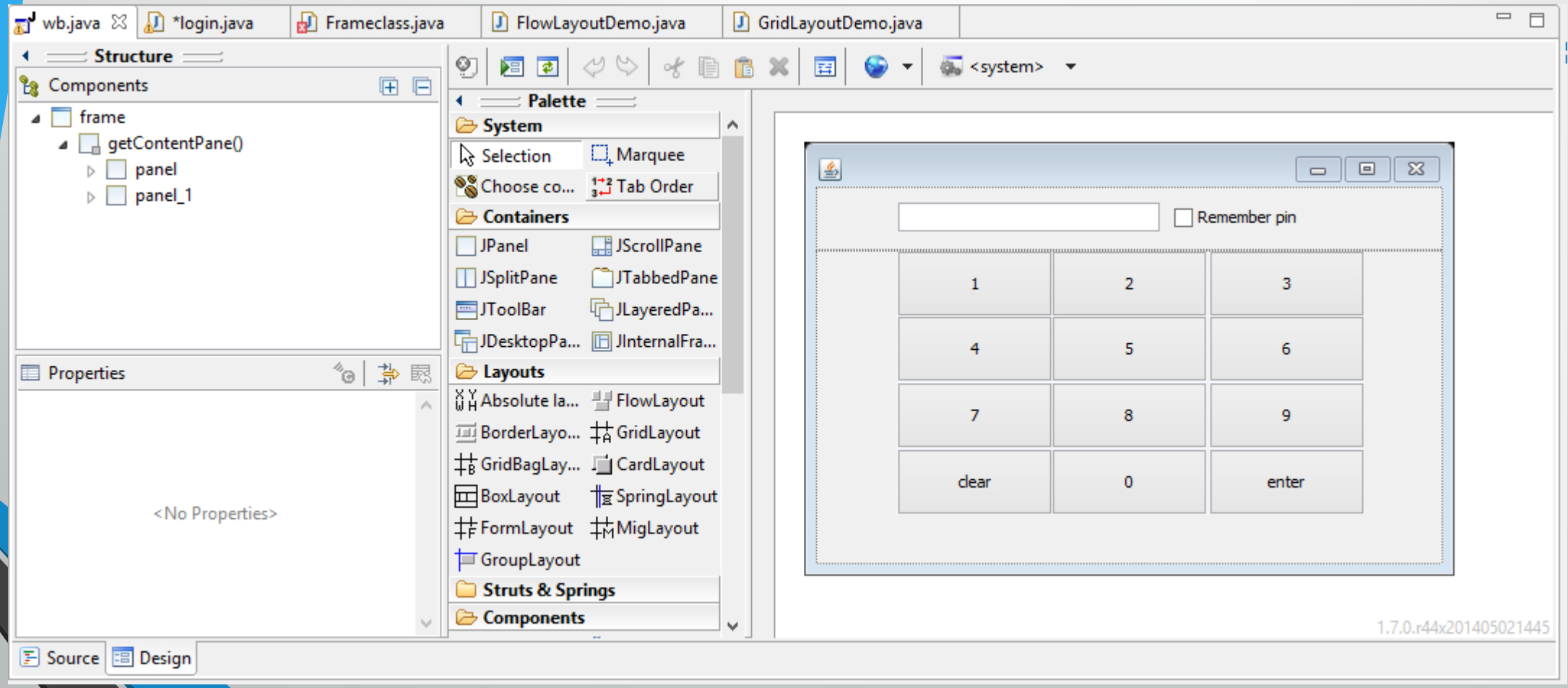
```
// Creating a JPanel panel and adding buttons
panel1 = new JPanel( new FlowLayout() );
panel1.add( new JButton( "Button 1" ) );
panel1.add( new JButton( "Button 2" ) );
panel1.add( new JButton( "Button 3" ) );

/** tempPanel - Temporary panel that will contain both panel1 and panel2*/
JPanel tempPanel = new JPanel( new BorderLayout() );
// Creating second panel and adding buttons
panel2 = new JPanel( new FlowLayout() );
panel2.add( new JButton( "Button 4" ) );
panel2.add( new JButton( "Button 5" ) );
panel2.add( new JButton( "Button 6" ) );
tempPanel.add( panel2, BorderLayout.CENTER );
tempPanel.add( panel1, BorderLayout.SOUTH );
```



WINDOWS BUILDER

There are many other ways to achieve correct alignment and layout.

- By using net beans IDE.
- By importing and linking a CSS stylesheet in java project.
- By using windows builder for eclipse.



The main task performed , using windows builder is correct alignment of all the items. The Items are instantiated as objects of their respective class automatically by the builder.

Event handlers are then coded for each item/component.

Event Handlers:

What is an event?

In computing, an event is an action or occurrence detected by the program. An event generally occurs when something changes within a graphical user interface. If a user clicks on a button, clicks on a combo box, or types characters into a text field, etc.. then an event will trigger.. There are many types of events.

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

Other event classes include MouseEvent, FocusEvent, KeyEvent WindowEvent etc.

What is a listener and a listener class?

A listener class is an interface or abstract class in event libraries. To implement this, a user made handler Class is created. The abstract methods of abstract listener class are called as listeners that actually Determine how will the system respond to an event.

Some common listener classes include: ActionHandler, InvocationHandler etc.

When a class implements a listener class, it overrides all its methods.(gives them a different definition)

```
private class theHandler implements ActionListener // this will be the class for handling all events
{
     public void actionPerformed(ActionEvent event)//overridden method of actionlistener class.
     {

     }

}
```

```
theHandler handler =new theHandler();
tf1.addActionListener(handler);
tf2.addActionListener(handler);
tf3.addActionListener(handler);
pf.addActionListener(handler);
```

Object.getsource();
object.getText();
Object.setText("");
Object.getActionCommand()

Thank you.