

Java Programming

Graphical User Interface (GUI)

Arash Habibi Lashkari

Ph.D. Candidate of UTM University

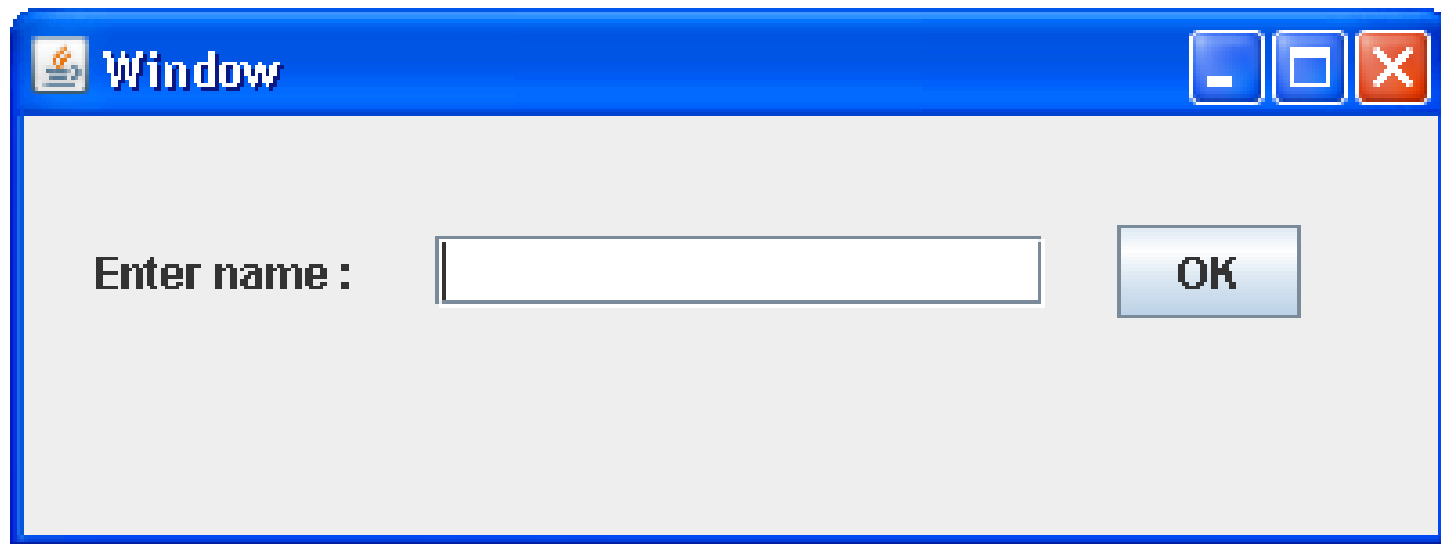
Kuala Lumpur, Malaysia

Overview

- Java classes to support programs with Graphical User Interface (GUI):
 - JFrame
 - JButton
 - JLabel
 - JTextField
 - JPasswordField
 - JCheckBox
 - JRadioButton
 - JList
 - JComboBox
 - JTextArea
 - JSlider
- GUI components are defined in **javax.swing**

JFrame

- A window where other GUI components will be displayed
- Example:



JFrame

- Constructor : **JFrame()**

- Member method:

setTitle(String s)

title of window

setSize(int w, int h)

width of window

height of window

x position of window

setLocation(int x, int y)

y position of window

setDefaultCloseOperation(JFrame.EXIT_ON_CLOS

E)

visible or not visible

setVisible(boolean v)

JFrame

```
import javax.swing.*;
public class TestJFrame
{
    public static void main (String[] args)
    {
        JFrame frame1 = new JFrame();
        frame1.setTitle("Window");
        frame1.setSize(200,150);
        frame1.setLocation(200, 400);

        frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame1.setVisible(true);
    }
}
```

JFrame



JButton

- Constructor: **JButton(String s)**

Label on button

- Example:

```
new JButton("OK");
```



JLabel

- Constructor: **JLabel()**
- Constructor: **JLabel(String s)**



Text on label

- Constructor: **JLabel(String s, int j)**



Text on label

alignment:



SwingConstants.LEFT
SwingConstants.CENTER
SwingConstants.RIGHT

JLabel

- Member method : **setText(String s)**
- Example :

```
new JLabel("Enter name : ");
```

or

```
JLabel lbl = new JLabel();  
lbl.setText("Enter name : ");
```



Enter name

JTextField

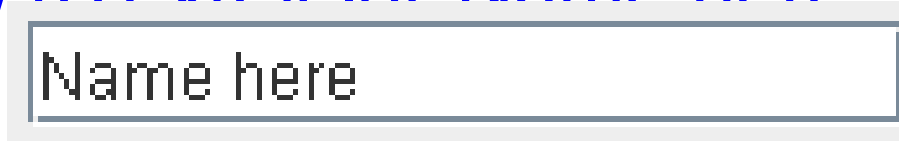
- Constructor: **JTextField(int j)**
- Example :

```
new JTextField(15);
```



- Constructor: **JTextField(String s, int j)**
- Example :

```
new JTextField("Name here" 15);
```



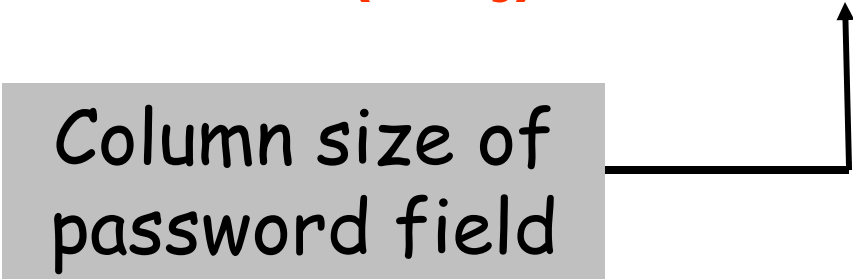
JTextField

- Member method:
- **setText(String s)**
 - to set the contents of text field to s.
- **getText()**
 - Return the contents of text field.
- **setColumns(int j)**
 - To set the size of text field to *j*.
- **setEditable(boolean editable)**
 - To set whether the contents of text field is editable. By default, a text field is editable.

JPasswordField

- constructor: **JPasswordField(int j)**

Column size of
password field



- Example :

```
new JPasswordField(10);
```



JCheckBox

□ Constructor: **JCheckBox(String s)**

□ Example :

```
new JCheckBox("Java");
```



label on check box



□ Constructor:

JCheckbox(String s, boolean status)

□ Example :

```
new JCheckBox("Java",true);
```



Initial status for check box



JCheckBox

- Member method: **isSelected()**
 - return status of check box, which is *true* (if selected) or *false* (if otherwise)

JRadioButton

□ Constructor: **JRadioButton(String s)**

□ Example :

```
new JRadioButton("Java");
```



label on check box



□ Constructor:

JRadioButton(String s, boolean status)

□ Example :

```
new JRadioButton("Java",true);
```



Initial status for check box



JRadioButton

- Member method: **isSelected()**
 - return status of radio button, which is *true* (if selected) or *false* (if otherwise)

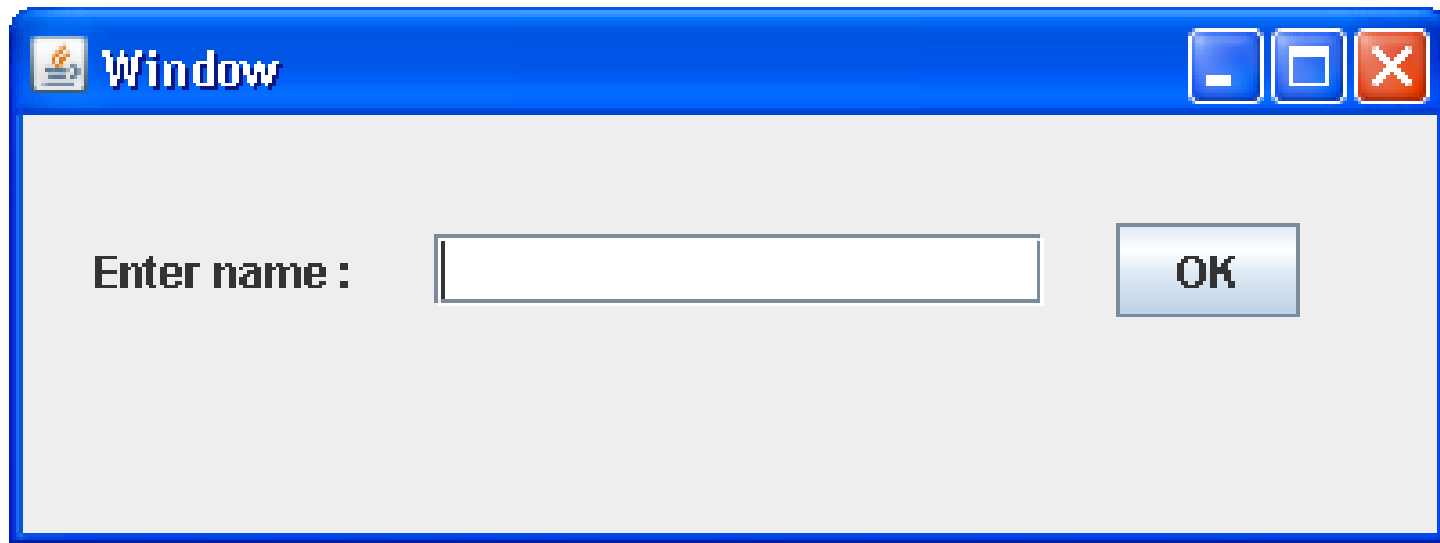
Adding GUI Components

- To add GUI components into JFrame:
 1. Set the layout of GUI components
 - by specifying a particular **Layout Manager**
 - Layout Manager is defined in **java.awt**
 2. Add components
 - using method **add()**

Adding GUI Components

```
public static void main (String[] args)
{
    JFrame frame1 = new JFrame();
    frame1.setTitle("Window");
    frame1.setSize(200,150);
    frame1.setLocation(200, 400);
    frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame1.setVisible(true);
    frame1.setLayout(new FlowLayout(FlowLayout.LEFT));
    JLabel label = new JLabel("Enter name : ");
    JTextField tf = new JTextField("Name here", 15);
    JButton bt = new JButton("OK");
    frame1.add(label);
    frame1.add(tf);
    frame1.add(bt);
}
```

Adding GUI Components



More GUIs

Back to JRadioButton ..

- **Example :**

```
:  
JRadioButton btn1, btn2, btn3;  
btn1= new JRadioButton("Windows", true);  
btn2 = new JRadioButton("Linux");  
btn3 = new JRadioButton("Solaris");  
frame1.add(btn1);  
frame1.add(btn2);  
frame1.add(btn3);  
:
```

Back to JRadioButton ..

- **Output:**



- **when other buttons are selected...**



ButtonGroup

- Object **ButtonGroup** needs to be created and linked to related radio buttons.
- Example:

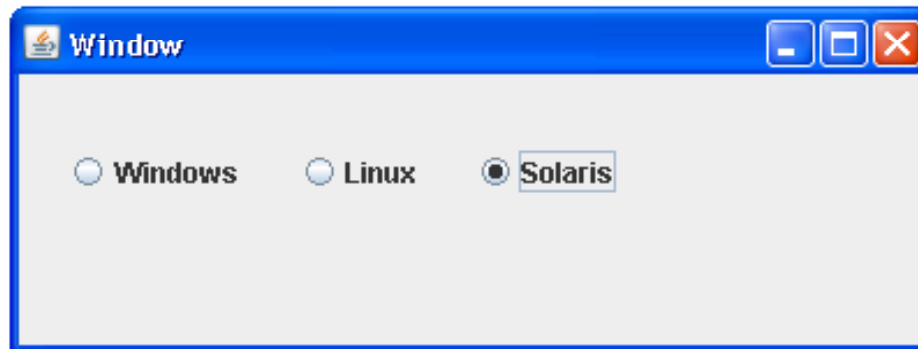
```
:  
JRadioButton btn1, btn2, btn3;  
btn1= new JRadioButton("Windows", true);  
btn2 = new JRadioButton("Linux");  
btn3 = new JRadioButton("Solaris");  
ButtonGroup gp = new ButtonGroup();  
gp.add(btn1);  
gp.add(btn2);  
gp.add(btn3);  
frame1.add(btn1);  
frame1.add(btn2);  
frame1.add(btn3);  
:
```

Using ButtonGroup ...

- **Output:**



- **when other buttons are selected...**



JList

- Constructor: **JList(Object[] list)**



Array containing
list of component
JList items

- Member method:

setVisibleRowCount(int c)

- set number of rows on list to be displayed without using scroll bar

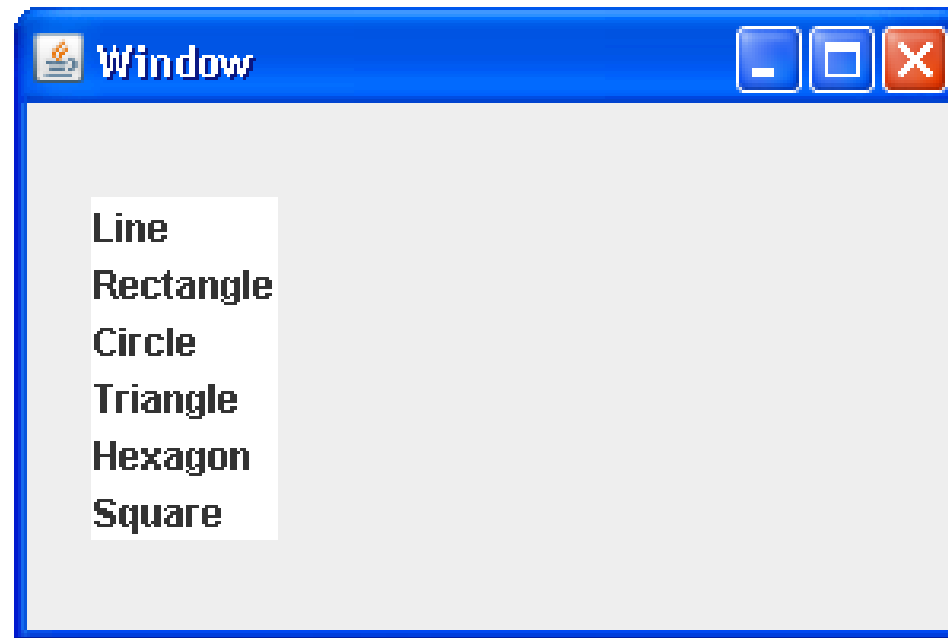
JList

- **Example :**

```
:  
String[] item = {"Line", "Rectangle", "Circle",  
                "Triangle", "Hexagon", "Square"};  
  
JList lst = new JList(item);  
lst.setVisibleRowCount(4);  
frame1.add(lst);  
:
```

JList

- Output:



- How many items are visible? Why?

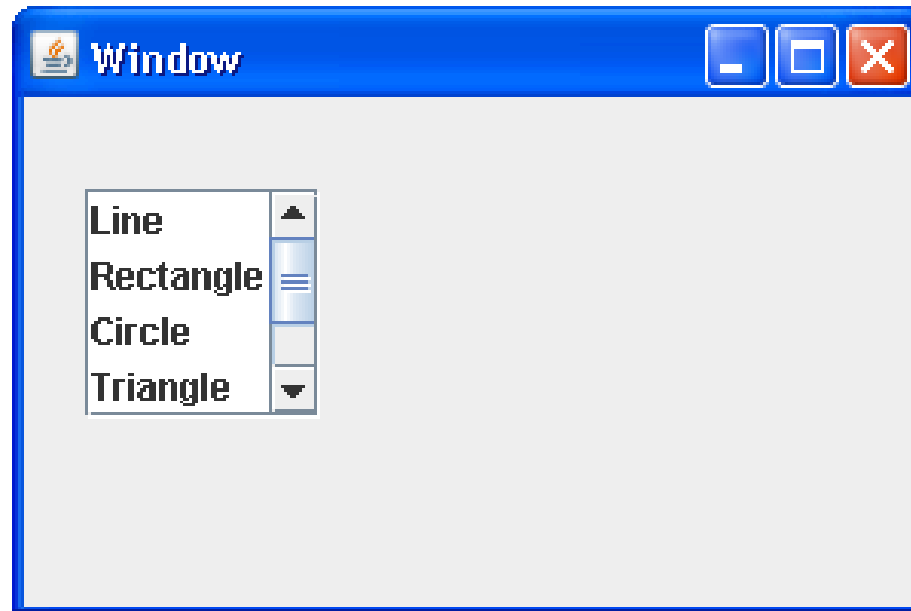
JScrollPane

- The object list should be linked to object `JScrollPane` before putting it into frame
- Example:

```
String[] item = {"Line", "Rectangle", "Circle",  
                "Triangle", "Hexagon", "Square"};  
  
JList lst = new JList(item);  
lst.setVisibleRowCount(4);  
JScrollPane jsp = new JScrollPane(lst);  
frame1.add(jsp);  
// frame1.add(lst);
```

JScrollPane

- Output :



- Only 4 items visible at a time, use scroll pane to see other items

Back to JList...

■ Member method:

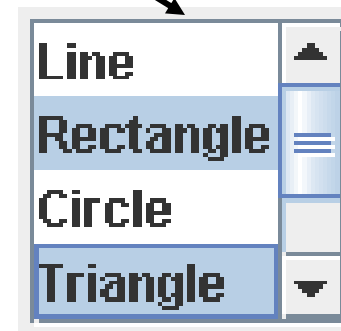
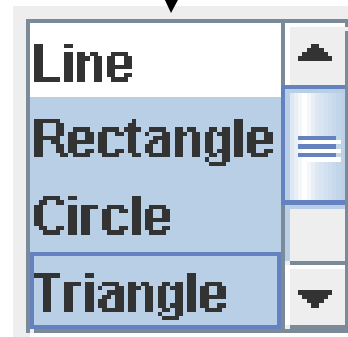
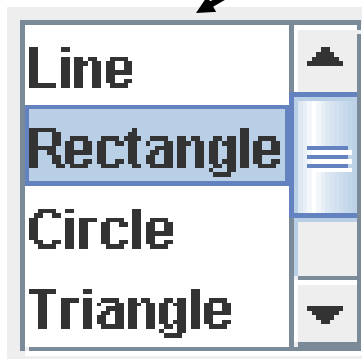
setSelectionMode

Set choice of mode

ListSelectionModel.SINGLE_SELECTION

ListSelectionModel.SINGLE_INTERVAL_SELECTION

ListSelectionModel.MULTIPLE_INTERVAL_SELECTION



JComboBox

□ Constructor:

JComboBox(Object[] list)

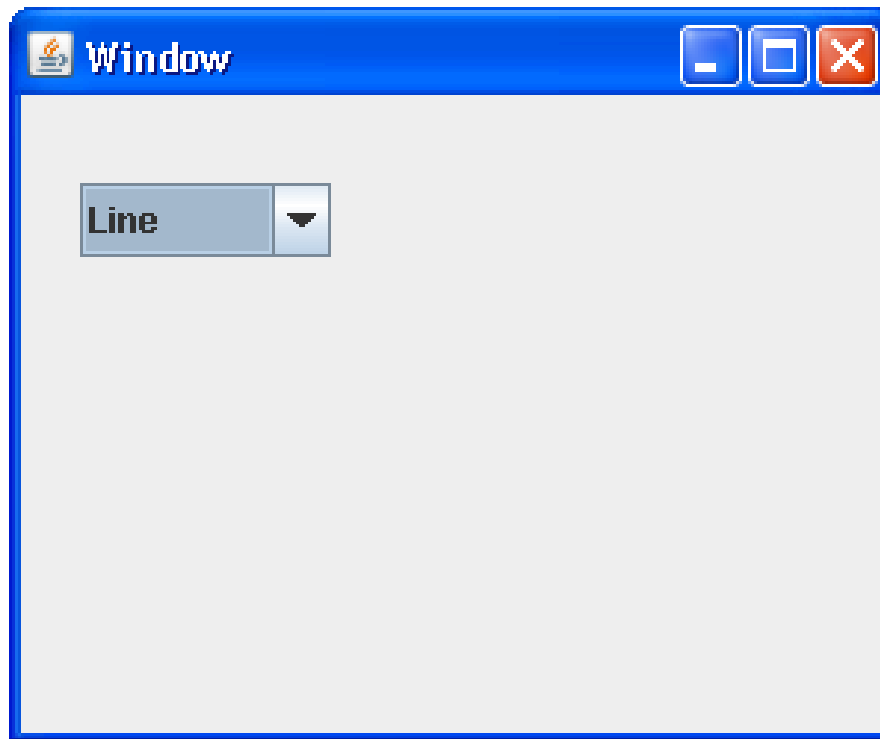
- Create object **JComboBox** which represents a combo box component whose item consists of objects in an array called *list*.

□ Example:

```
:  
String[] item = {"Line", "Rectangle", "Circle",  
                "Triangle", "Hexagon", "Square"};  
JComboBox cb = new JComboBox(item);  
frame1.add(cb);  
:
```

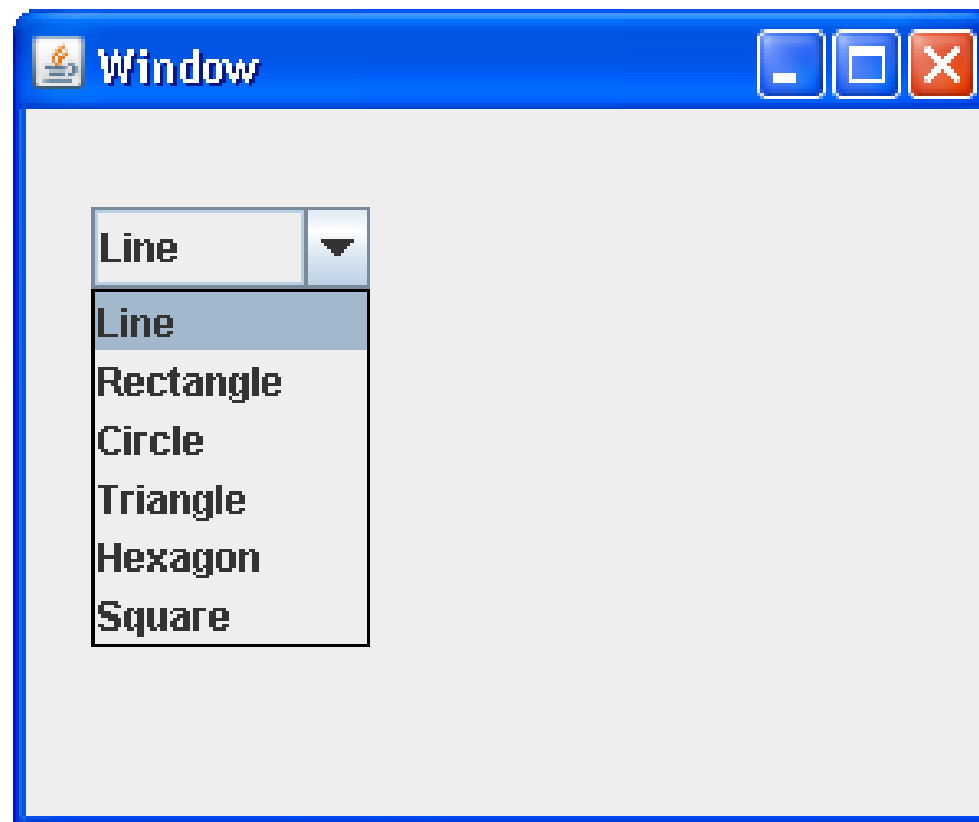
JComboBox

- Output:



JComboBox


- After clicking the down arrow:



JTextArea


□ constructor:

JTextArea(int br, int lj)



Number of rows and
columns of text area

JTextArea(String tks, int br, int lj)



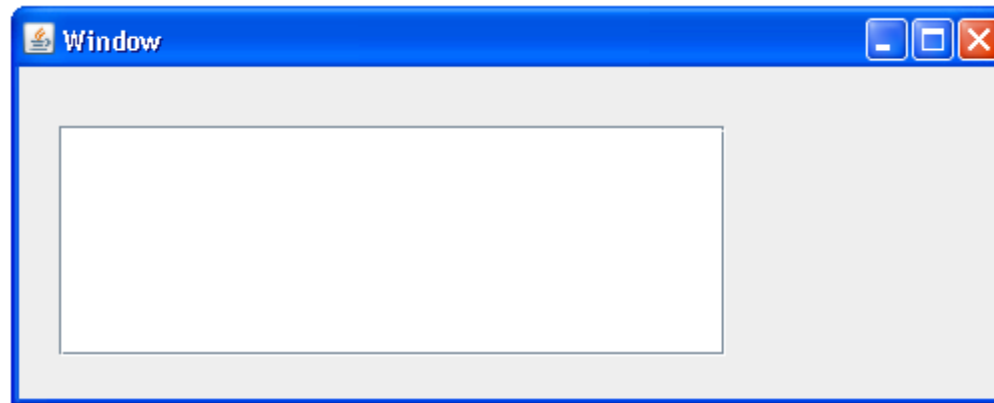
Initial text in
text area

JTextArea

□ Example:

```
:  
JTextArea ta = new JTextArea(7, 30);  
JScrollPane jsp = new JScrollPane(ta);  
frame1.add(jsp);  
:
```

□ Output:

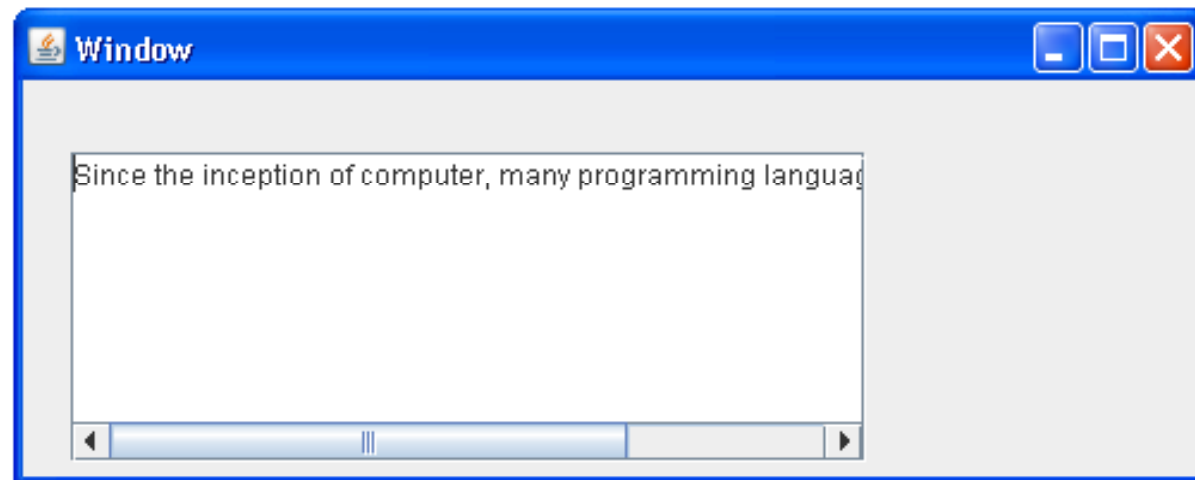


JTextArea

□ Example:

```
:  
    JTextArea ta = new JTextArea("Since the inception of  
        computer, many programming languages have        been  
created.",7, 30);  
    JScrollPane jsp = new JScrollPane(ta);  
    frame1.add(jsp);  
:
```

□ Output:



JSlider

□ Constructor:

JSlider()

Example:

```
new JSlider();
```

Create default object JSlider :
horizontal
range 0..100
Initial value 50.

JSlider(int direction)



JSlider.HORIZONTAL
JSlider.VERTICAL

Example:

```
new JSlider(JSlider.HORIZONTAL);
```

JSlider

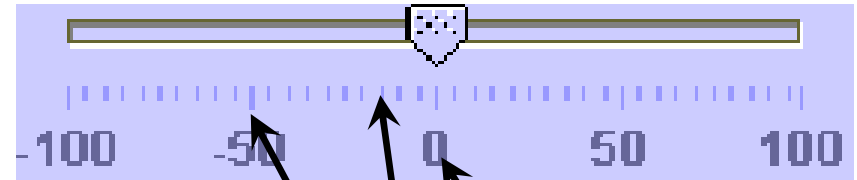
JSlider(int dir, int min, int max, int n)

- Create object `JSlider` with range *min..max* and its initial value *n*.

□ Example:

```
new JSlider(JSlider.HORIZONTAL, -100, 100, 0);
```

JSlider



Member method:

void setMajorTickSpacing(int)

- Set the spacing of major tick marks

void setMinorTickSpacing(int)

- Set the spacing of minor tick marks

void setPaintTicks(boolean)

- true – to display major and minor tick marks

void setPaintLabels(boolean)

- true – to display label for major tick marks

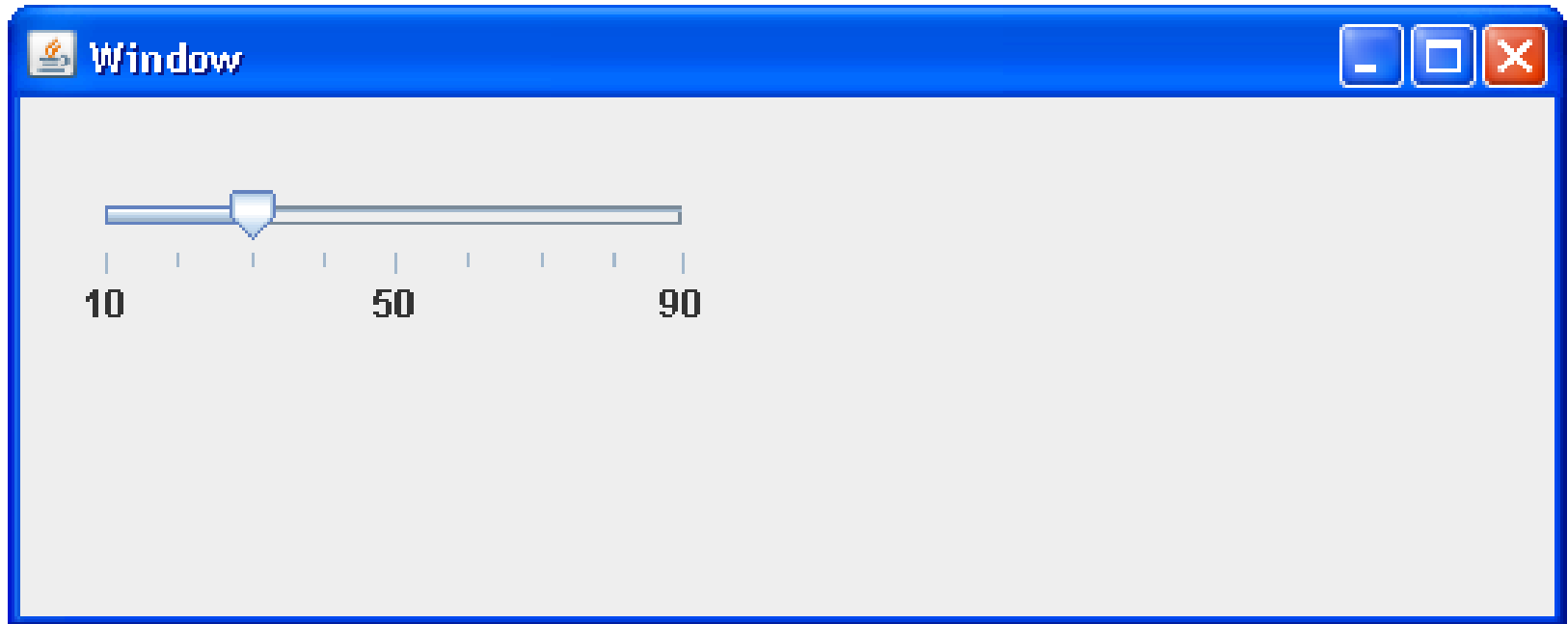
JSlider

- Example:

```
:  
  JSlider slider = new JSlider(JSlider.HORIZONTAL,  
    10, 90, 30);  
  slider.setMajorTickSpacing(40);  
  slider.setMinorTickSpacing(10);  
  slider.setPaintLabels(true);  
  slider.setPaintTicks(true);  
  frame1.add(slider);  
:
```

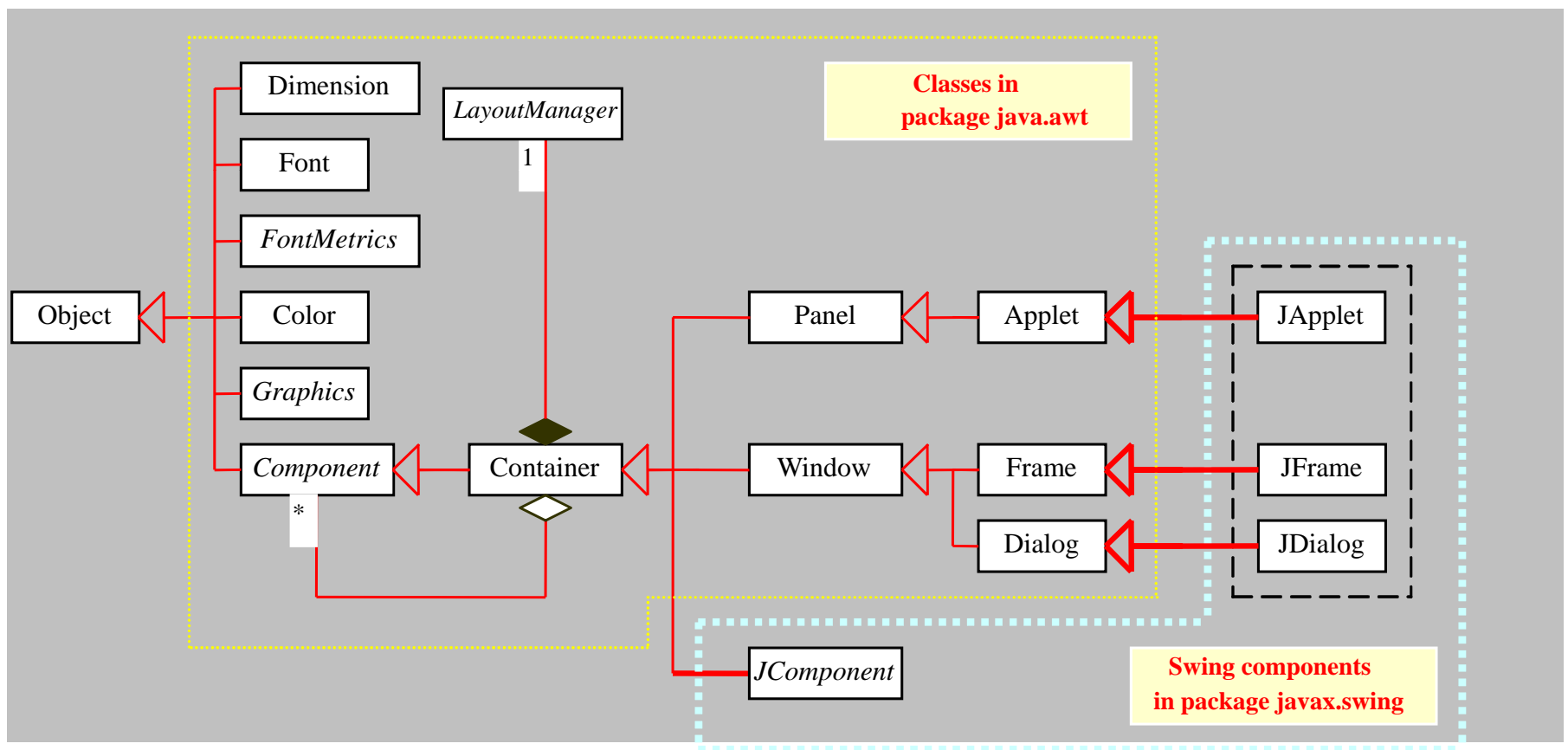

JSlider

- Output:

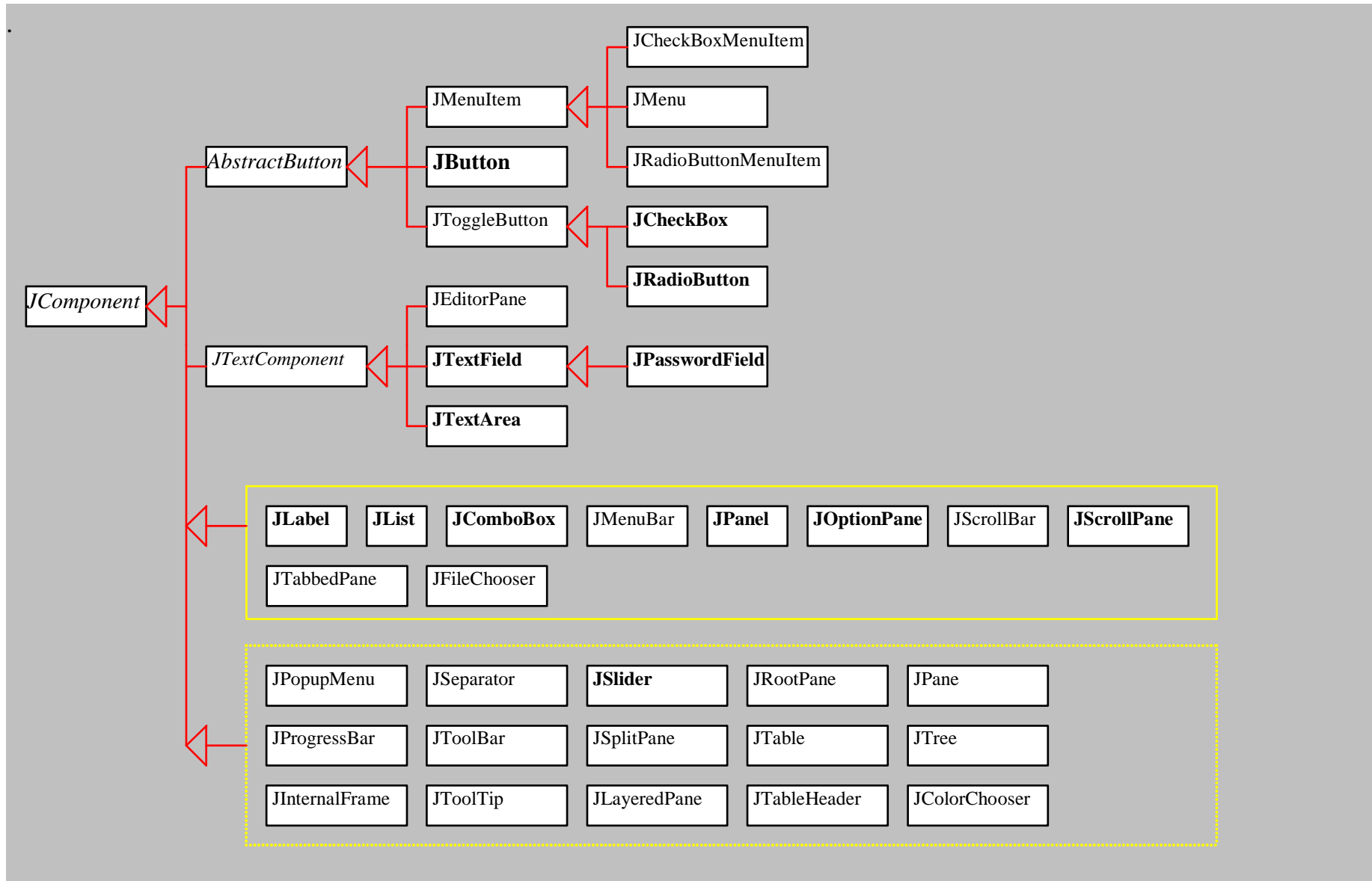


Hierarchy of Swing Components

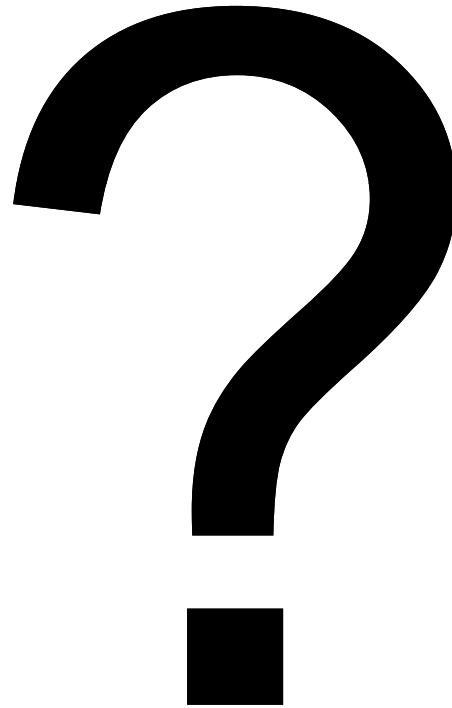
Basic components of GUI Swing inherit class JComponent



JComponent



Questions



THANK YOU

Arash Habibi Lashkari

PHD. Candidate of UTM

Kuala Lumpur, Malaysia

Feb, 2010

THE END