Core Java
Unit-IV
Topic:
GUI Development using AWT and Swing

What is a User Interface (UI)

What is a UI?

- Means by which software end-users interact with your software (system)

Types of UI

- Command line interface (CLI) (e.g. DOS, Unix – older systems)
- Graphical User interface (GUI) (e.g. windows, Mac)
- Natural Language interface (e.g. Android contact search, voice dial, Google voice search)

What is a GUI ?

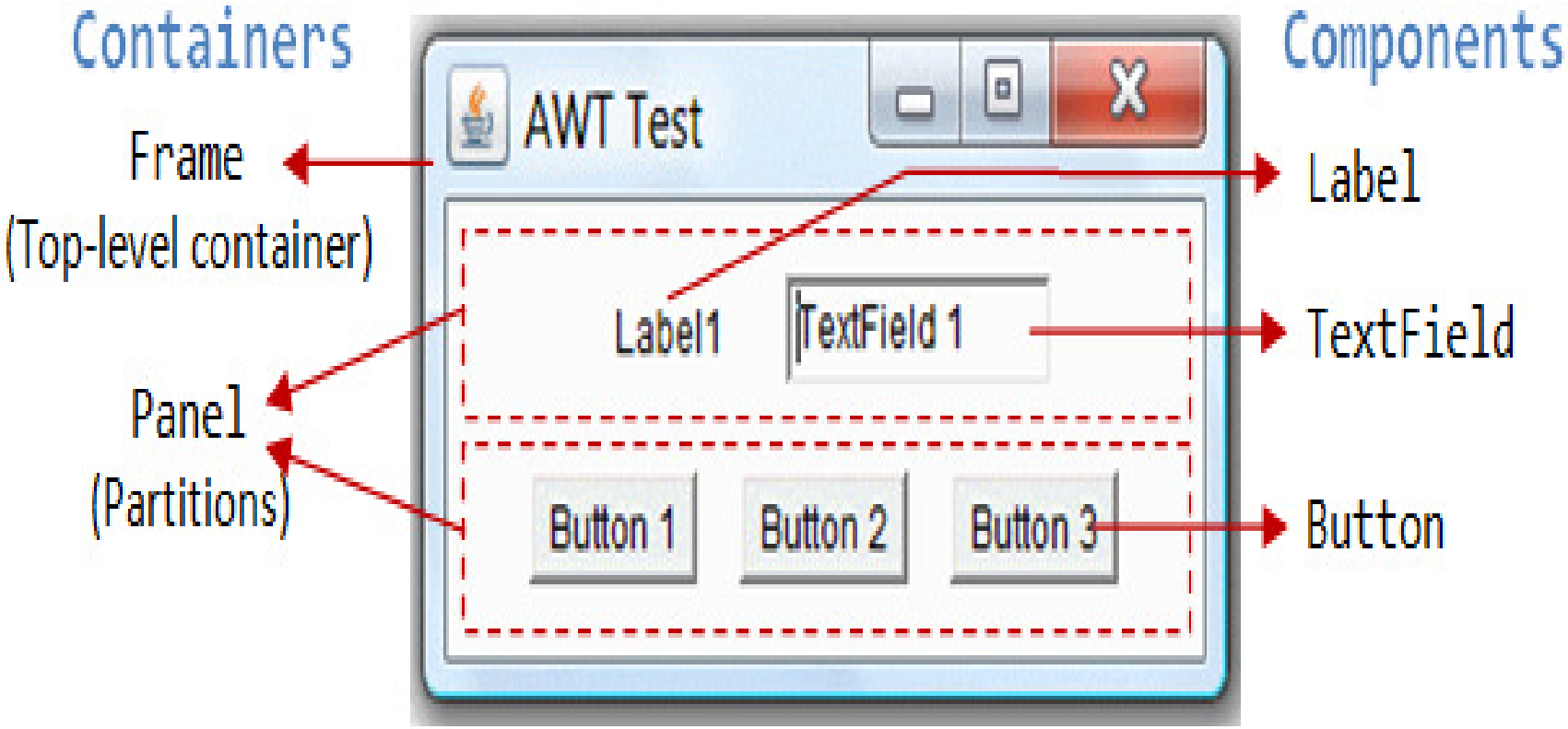
- What is Graphical user interface?
- Window based user interface (as against command based user interface /console based – black and white)
- GUI controls – window, button, label, textbox, text area, combo-box, list box, etc..

java.awt (Abstract Window Toolkit)

- Introduced in JDK 1.0 - Provides a platform-independent and device-independent GUI
- Graphics programs that runs on all platforms, such as Windows, Mac, and Linux

1. GUI Component classes - for creating GUI controls like button, text field etc.
2. GUI Container classes (different types of windows such as Frame, Panel, Dialog etc.)
3. Layout managers (to adjust/arrange GUI controls in containers according to window size or screen resolution)

java.awt (Abstract Window Toolkit)



java.awt (Abstract Window Toolkit)

- Event-handling model (e.g. handling button click)
- Graphics and imaging classes - drawing different shapes – line, circle etc., using colors and fonts

Java - GUI development packages

javax.swing (added in JDK 1.1 as part of JFC-Java Foundation Classes)

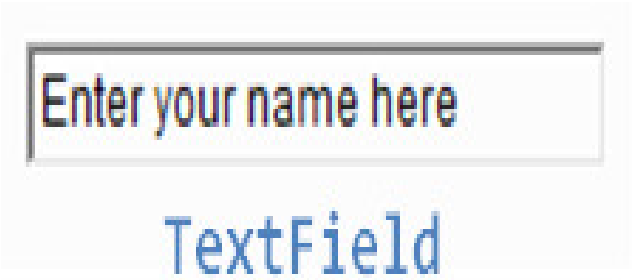
- Lightweight and faster/efficient than AWT
- Extends the classes in java.awt package
  E.g. javax.swing.JTextField → java.awt.TextField
- New methods and capabilities added in AWT controls
- javax.swing.JButton - both image and text - sorting, printing, and drag and drop capability
- New GUI controls – PasswordField, JTable and tree

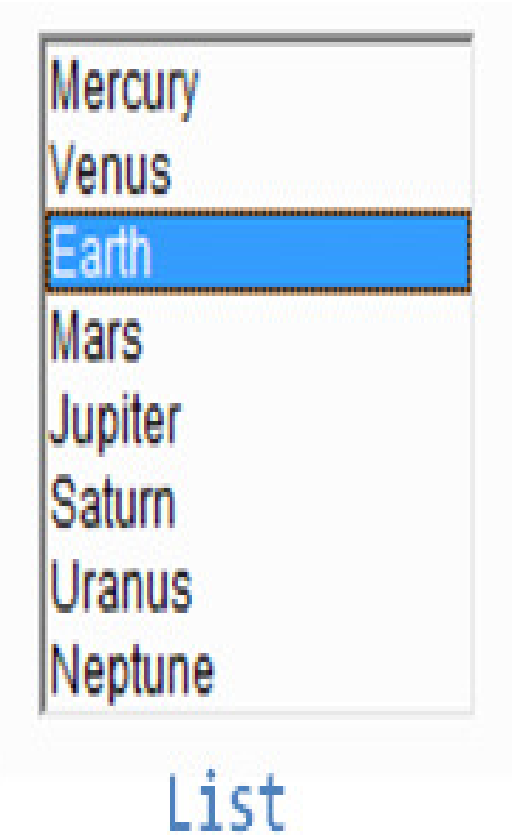
Java - GUI development packages

javax.swing

- not implemented by native (platform-specific code), choice of look and feel – windows or Java look)

AWT - Class Hierarchy





Using Eclipse – WindowBuilder plugin

- Allows rapid development of GUI
- Drag and drop GUI controls from toolbox
- Using javax.swing classes e.g. JTextField, JFrame
- Swing code is auto-generated

- Steps –
- Right-click on src
- Create new → WindowBuilder → Swing Designer → JFrame

Using Eclipse – WindowBuilder plugin

- How to open a normal Java program (.java file) with handwritten GUI code – open with WindowBuilder Plugin (for getting Design View)
- Steps –
- Open Package Explorer
- Right-click on Java file – Open With – WindowBuilder editor

Eclipse WindowBuilder plugin download website

http://www.eclipse.org/windowbuilder/download.php

- Copy the release version link of Eclipse 4.4 (Luna)
- http://download.eclipse.org/windowbuilder/WB/release/R201506241200-1/4.4/

How to install WindowBuilder plugin

eclipse

Google™ Custom Search

GETTING STARTED MEMBERS PROJECTS MORE

DONATE

HOME / PROJECTS / WINDOWBUILDER / INSTALLING WINDOWBUILDER PRO

MyProject

- Download
- Documentation
- Support

Installing WindowBuilder Pro

All downloads are provided under the terms and conditions of the **Eclipse Foundation Software User Agreement** unless otherwise specified.

Develop Java graphical user interfaces in minutes for Swing, SWT, RCP and XWT with WindowBuilder Pro's WYSIWYG, drag-and-drop interface. Use wizards, editors and intelligent layout assist to automatically generate clean Java code, with the visual design and source always in sync.

These instructions assume that you have already installed some flavor of Eclipse. If you have not, Eclipse can be downloaded from **http://www.eclipse.org/downloads/**. Instructions and system requirements for installing WindowBuilder can be found **here**.

Update Sites

Eclipse Version	Release Version		Integration Version	
	Update Site	Zipped Update Site	Update Site	Zipped Update Site
4.5 (Mars)	link	link (MD5 Hash)	link	link (MD5 Hash)
4.4 (Luna)	link	link (MD5 Hash)	link	link (MD5 Hash)
4.3 (Kepler)	link	link (MD5 Hash)		
4.2 (Juno)	link	link (MD5 Hash)		
3.8 (Juno)	link	link (MD5 Hash)		

How to install WindowBuilder plugin

Eclipse Luna IDE (4.4 version) - WindowBuilder plugin installation using below steps:

1) Open Eclipse Luna IDE
2) Go to Help menu – Install new software
3) In work with (textbox) – enter the below link and press enter (to install the plugin)

http://download.eclipse.org/windowbuilder/WB/release/R20150 6241200-1/4.4/

Internet must be working for the plugin to get installed

Manuall install of WindowBuilder plugin

Download the Eclipse Luna IDE (4.4 version) - WindowBuilder plugin (Integrated version - zipped update site/folder in local computer): WB_v1.8.0_UpdateSite_for_Eclipse 4.4

1) Unzip the above folder
2) Copy the plugins and features folders and PASTE IT in Eclipse installation folder in location – C:\Program Files\eclipse

Using Eclipse – WindowBuilder plugin

Java - gui/src/CalculatorFrame.java - Eclipse

File Edit Source Refactor Navigate Search Project Run WTP Tests Releng Window Help

CalculatorFrame.java

Structure

Components

(javax.swing.JFrame) - "Cal

contentPane

lblCalculatorOperati

lblNumber - "Numbe

textField

lblNewLabel - "Num

textField_1

lblResult - "Result"

textField_2

Properties

<No Properties>

Calculator

Calculator Operations

Number1

Number2

Result

ADD

SUBTRACT

Palette

System

Selection Marquee

Choose c... Tab Order

Containers

JPanel JScrollPane

JSplitPane JTabbedP...

JToolBar JLayeredP...

JDesktopP... JInternalFr...

Layouts

Absolute l... FlowLayout

BorderLay... GridLayout

GridBagL... CardLayout

BoxLayout SpringLay...

FormLayo... MigLayout

GroupLay...

Struts & Springs

Components

JLabel JTextField

JComboB... JButton

WindowBuilder 1.0.0.r36x201106161111

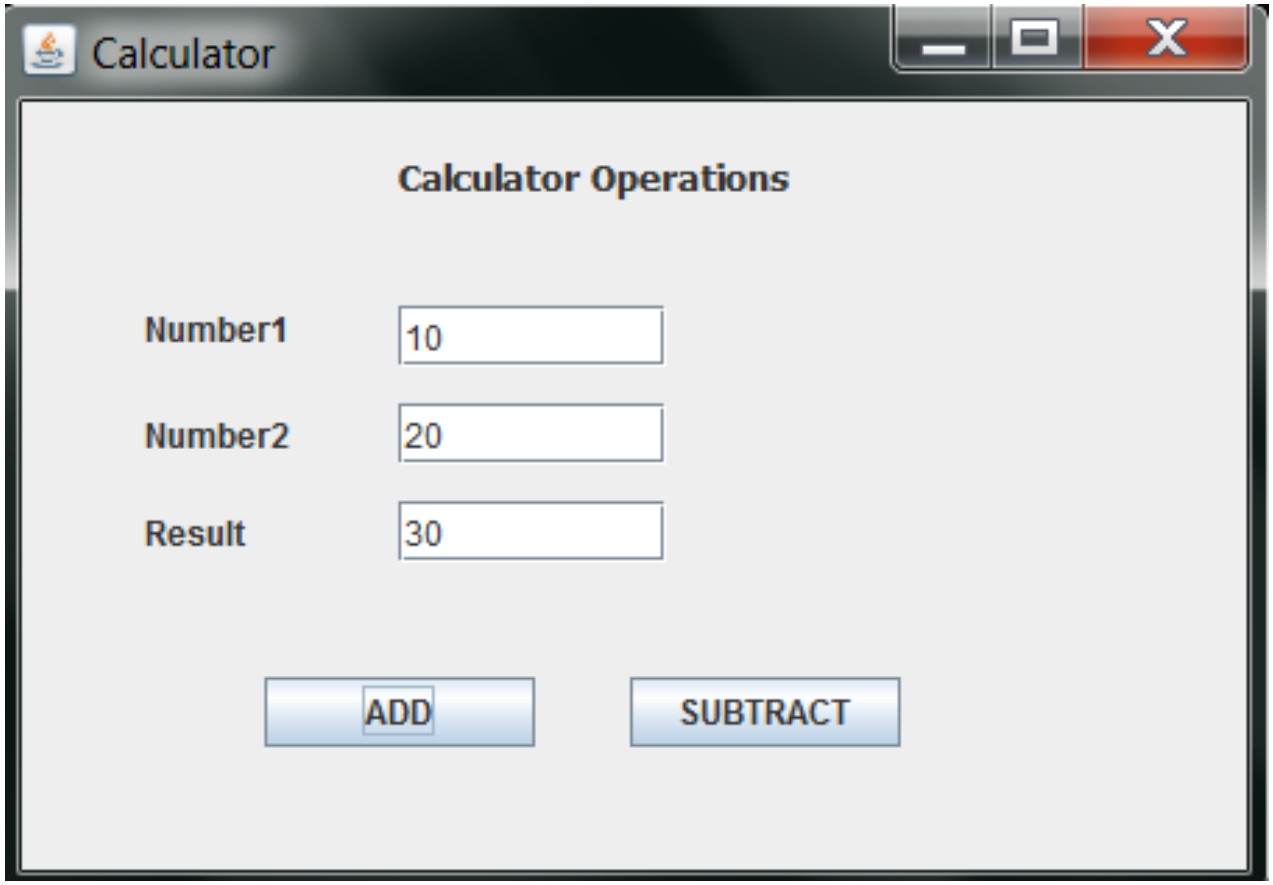
Source Design

Writable Smart Insert 19 : 18

10:44 AM 3/8/2015

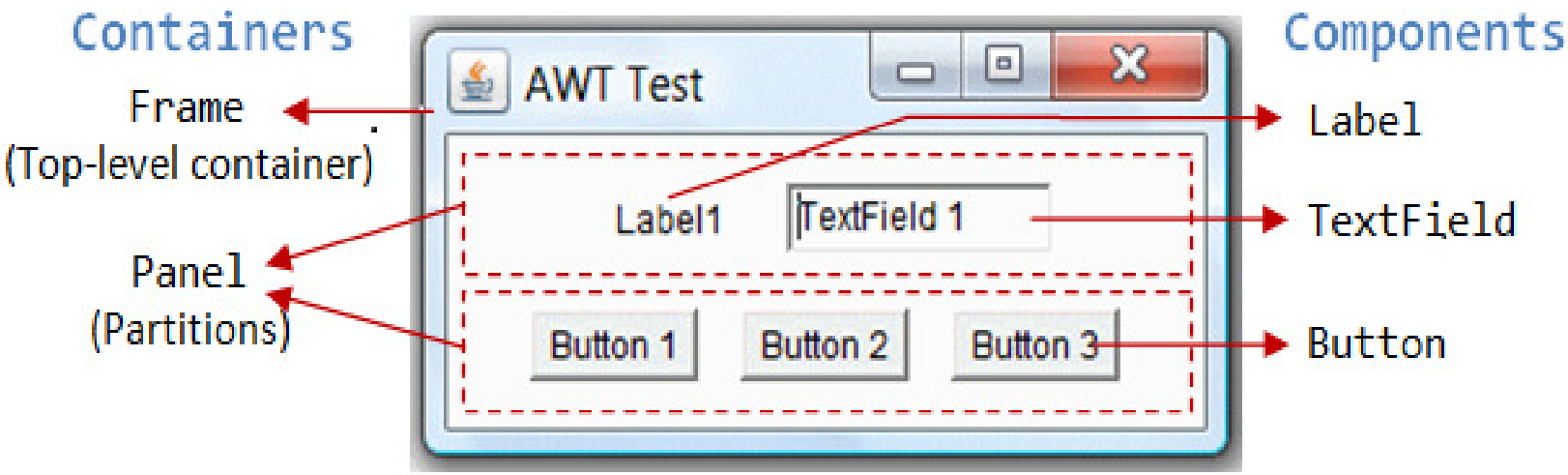
AWT - Windows based program

- → AddTwoNumberFrame.java



Components and Containers

- 2 types of GUI elements
- Component: GUI controls/widgets (such as Button, Label, and TextField, TextArea)
- Container: such as Frame, Panel and Applet are used to hold components



Add components to Containers

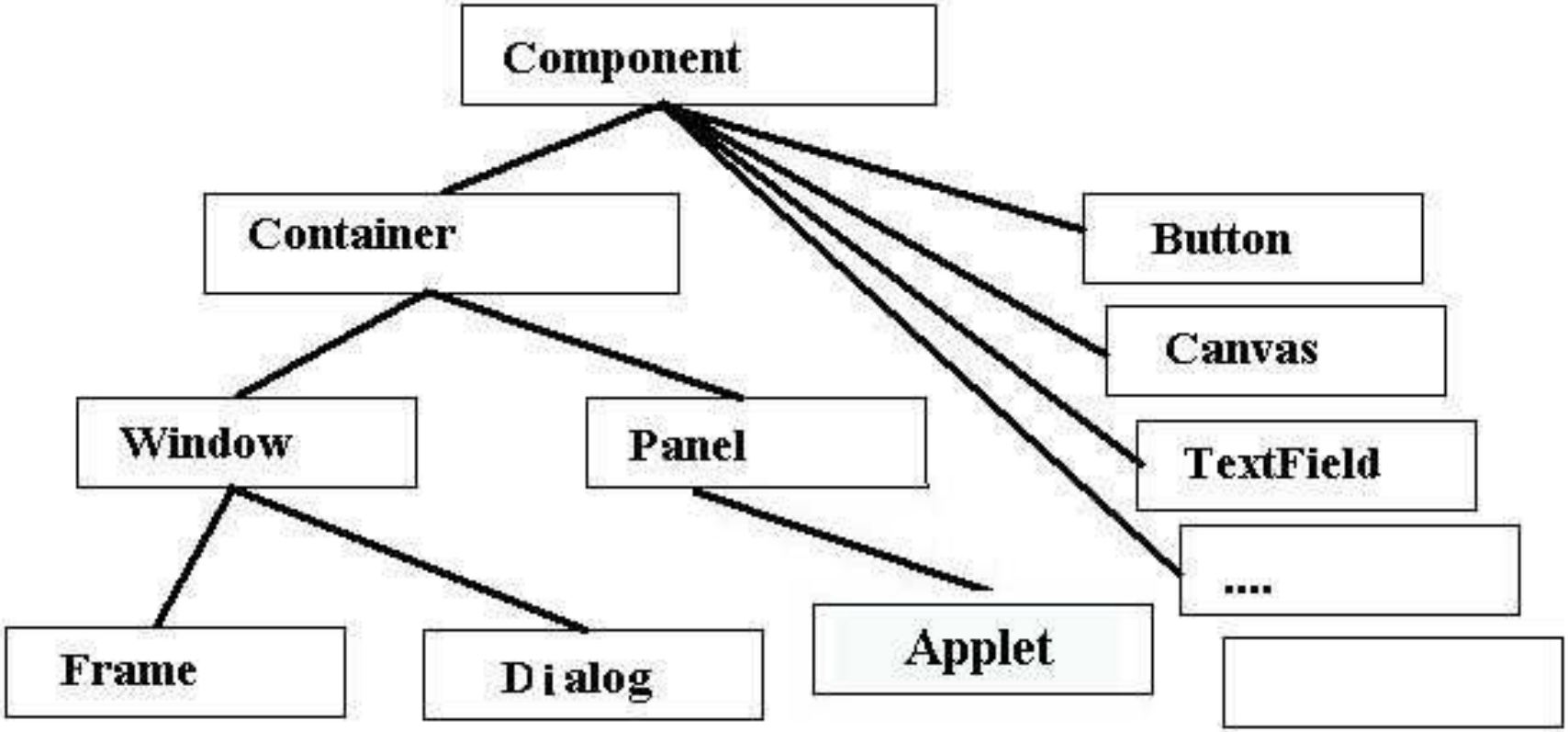
- In a GUI program, a component must be kept in a container.
- Every container has a method called add(Component c).
- Frame panel = new Frame(); // Frame is a Container

  Frame frame = new Frame("Title of frame");

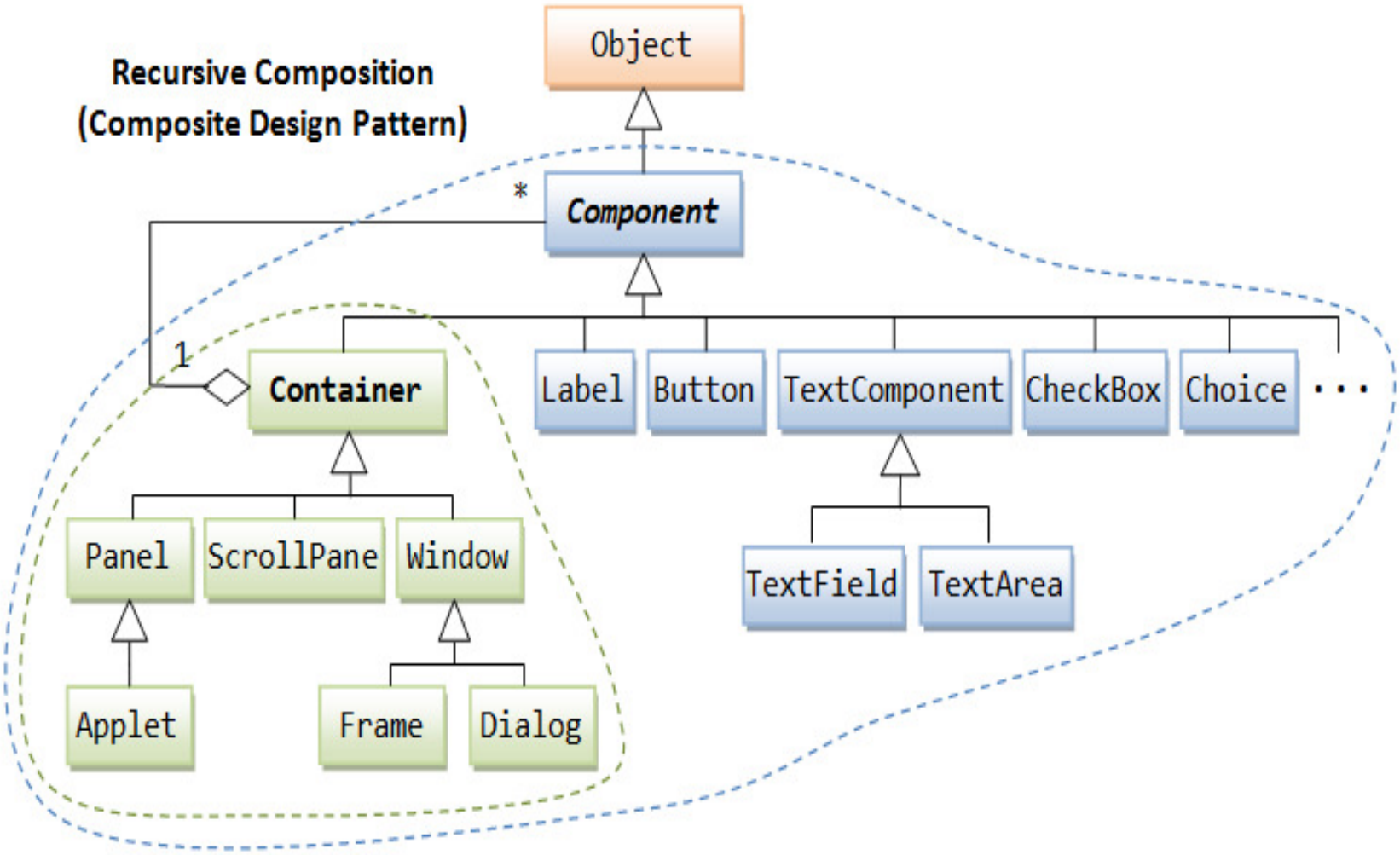
  frame.add(btn);

  // The Panel Container adds a Button Component

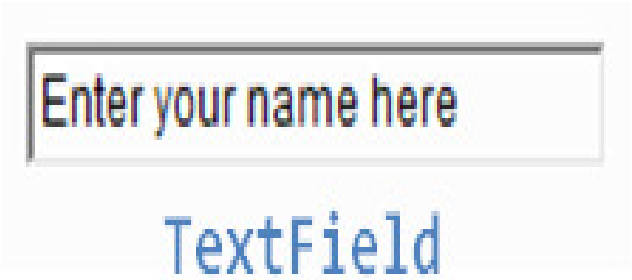
AWT - Class Hierarchy

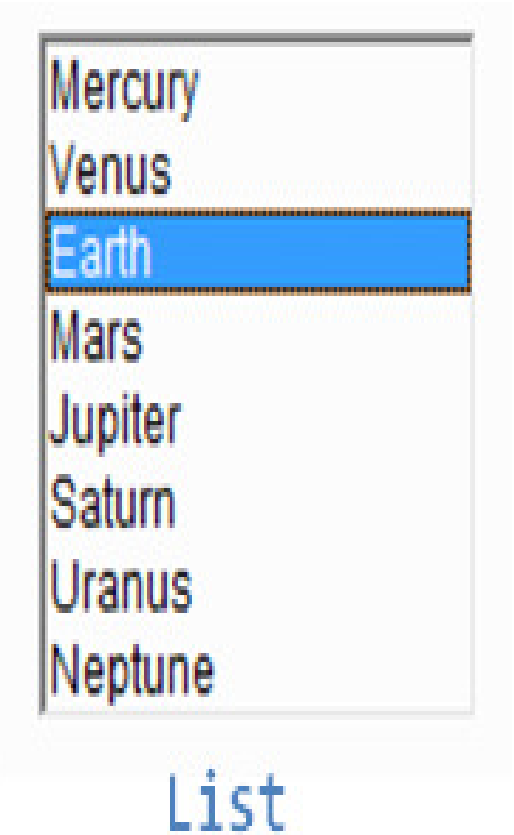


AWT - Class Hierarchy

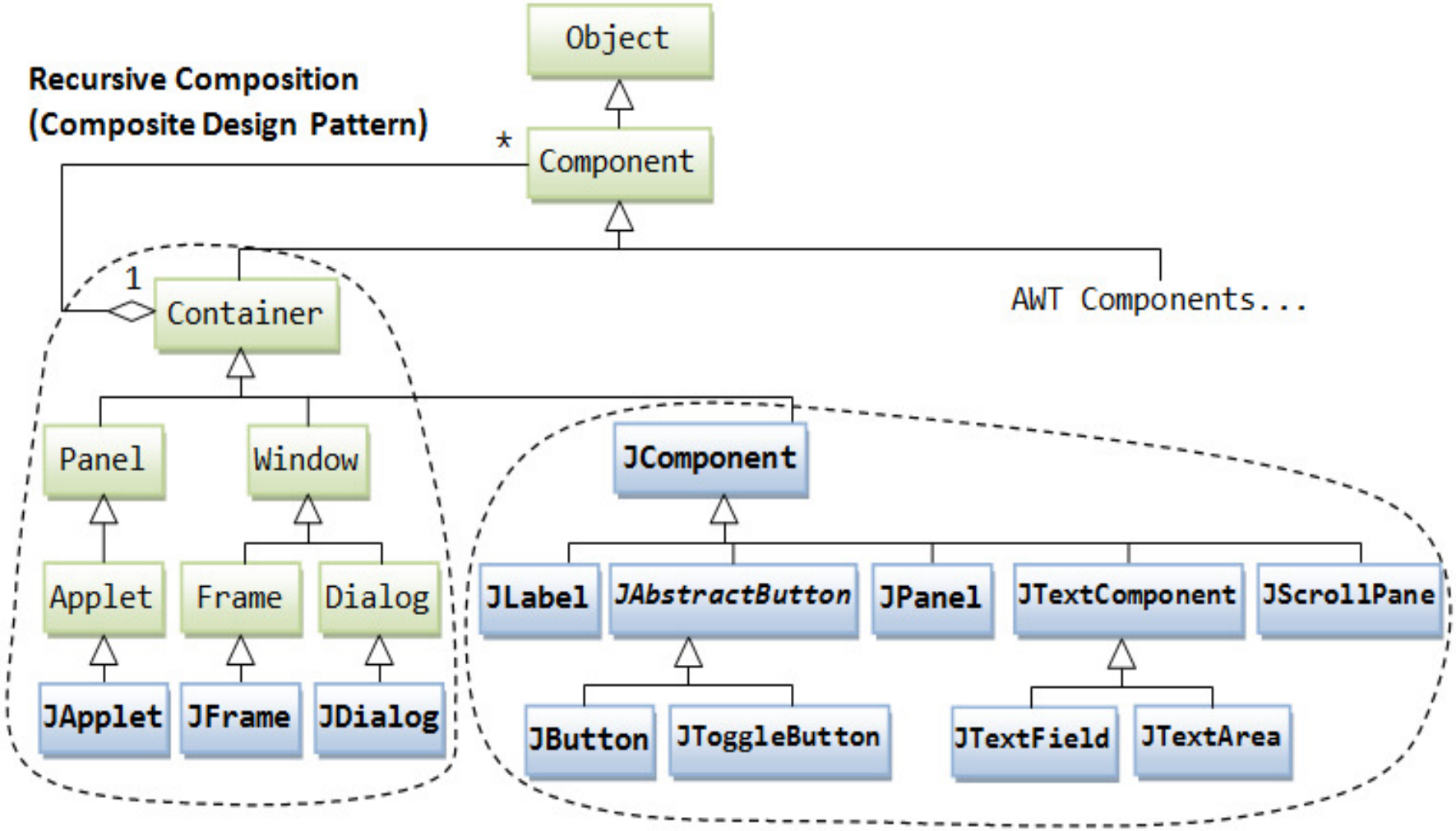


AWT - Class Hierarchy

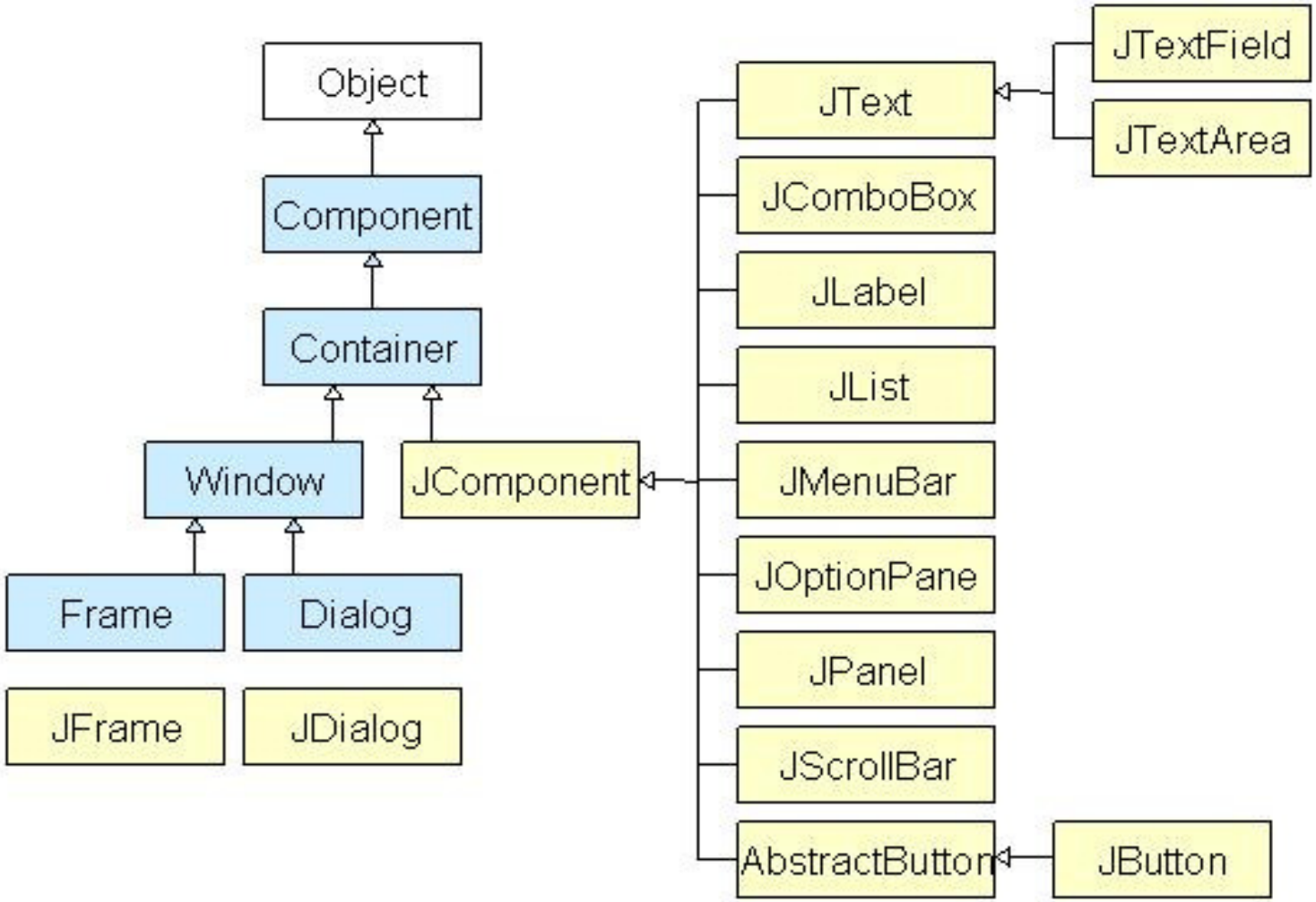




Swing - Class Hierarchy

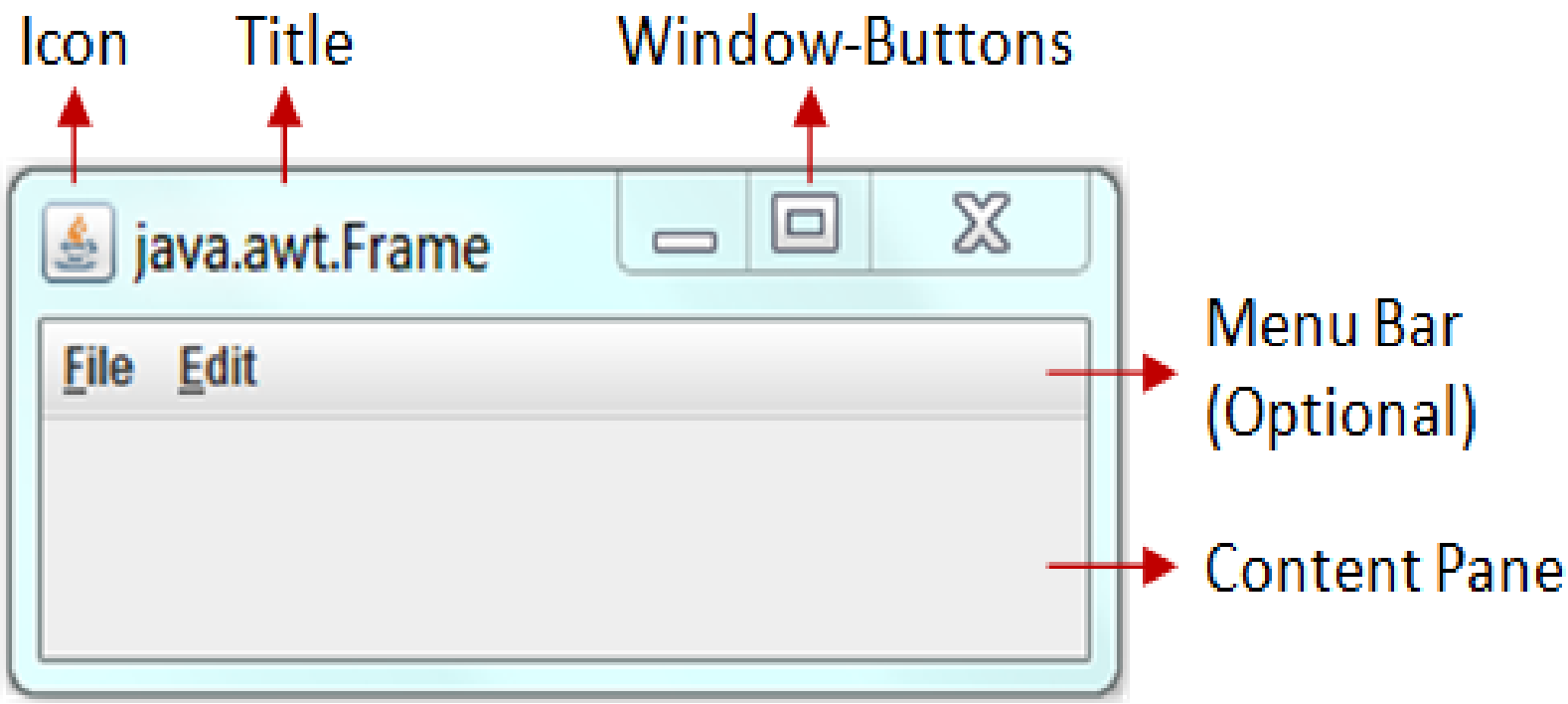


Swing Class Hierarchy



Frame

- **Frame -** represents a normal "window" with **title bar, icon, borders, and window buttons**.
- Used to create top-level window in a Application
- Can have menu bar



Methods of Frame Class

- Used to create top-level window in a Application
- Constructor:
- Frame frame = new Frame(“Title of frame”);
- Frame frame = new Frame(); //without title

- Methods:

1. frame.setTitle("Addition of 2 numbers in Window");
2. frame.setSize(int w, int h); //width, height in pixels
3. Dimension newSize=new Dimension(400,200);
   frame.setSize (newSize);

Methods of Frame Class

4. After you create Frame window – it is not visible.

   void setVisible(true);

5. frame.setBackground(Color.cyan);
6. void setIconImage(Image image) - Sets the image icon for this window.
7. void setMenuBar(MenuBar mb) - Sets the menu bar for this frame.
8. void setResizable(boolean resizable) - whether frame is resizable or not.

Event Handling for Frame

- Frame.addWindowListener(windowListener);

- class FrameEventListener extends WindowAdapter{

```
    public void windowClosing(WindowEvent e) {
            frame.setVisible(false);
            System.exit(0);     //close the application
    }
};
```

Creating Frame object

- 1) Define class – extends – java.awt.Frame class
- 2) Create label, textbox, button objects
- 3) add( ) to Frame – after specifying layout
- 4) In main( ) method, create object of Frame class

  AddTwoNumberFrame frame = new AddTwoNumberFrame();
- frame.setTitle("Addition of 2 numbers in Window");
- frame.setSize(400,300);
- frame.setVisible(true);

Creating Frame window based program

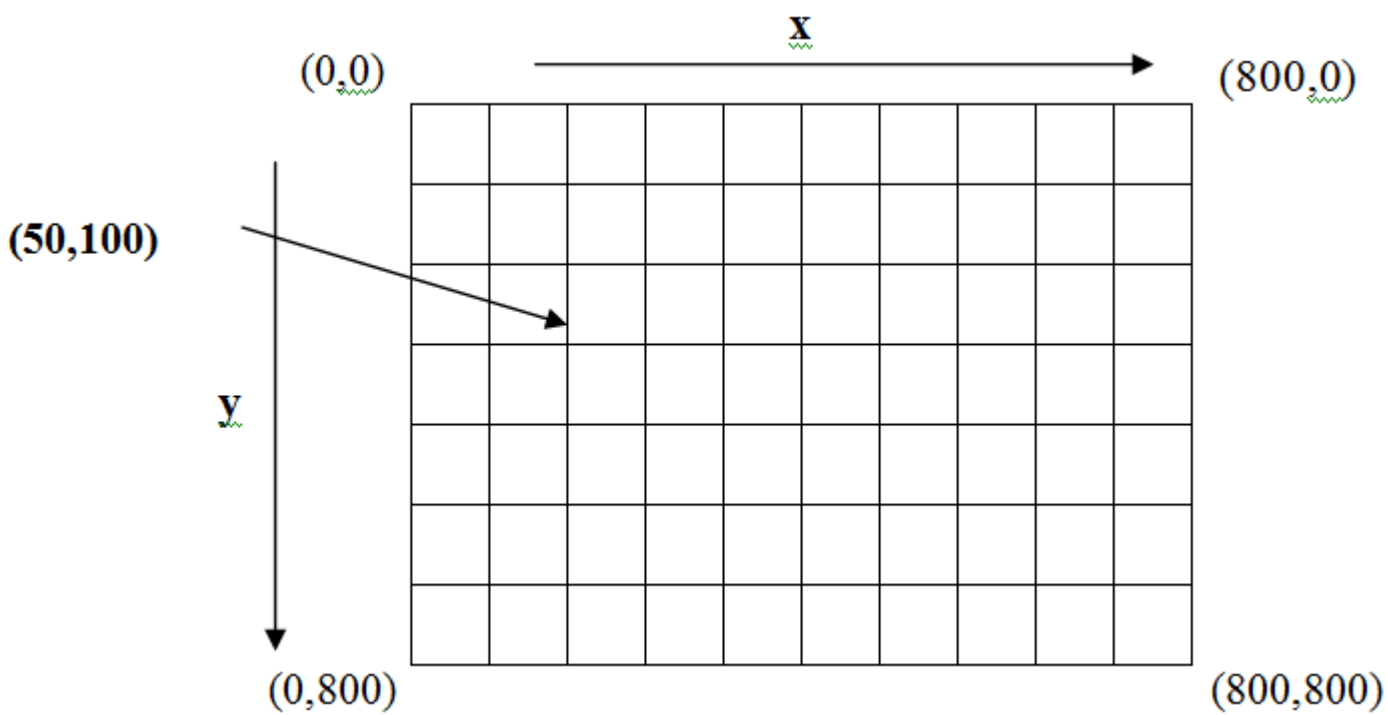
```
import java.awt.Frame;
// A GUI program is written as a subclass of Frame –
// the top-level container
public class MyGUIProgram extends Frame {
    // Constructor to setup the GUI components
    public MyGUIProgram( ) {
     ......
    }
    // Other methods ...... ......
```

Creating Frame window based program

```
// The entry main() method
public static void main(String[] args) {
  // Invoke the constructor (to setup the GUI) by
  //allocating an instance
        new MyGUIProgram();
  }
}
```

Coordinate System in Java

- Origin (0,0) in upper-left corner.
- (x,y) = x is distance from left , Y=distance from top
- Value of coordinates x and y are in pixels.



JRadioButton – how to allow selection of only one

```
//Allow Selection Of Only One Radio Button
//Create Group

ButtonGroup group = new ButtonGroup();
group.add(maleJRadioButton);
group.add(femaleJRadioButton);
```

Note: In AWT class - RadioButton, CheckboxGroup is used to create a group

→ Refer StudentRegistrationFrame.java

How to create Frame with same as size of screen

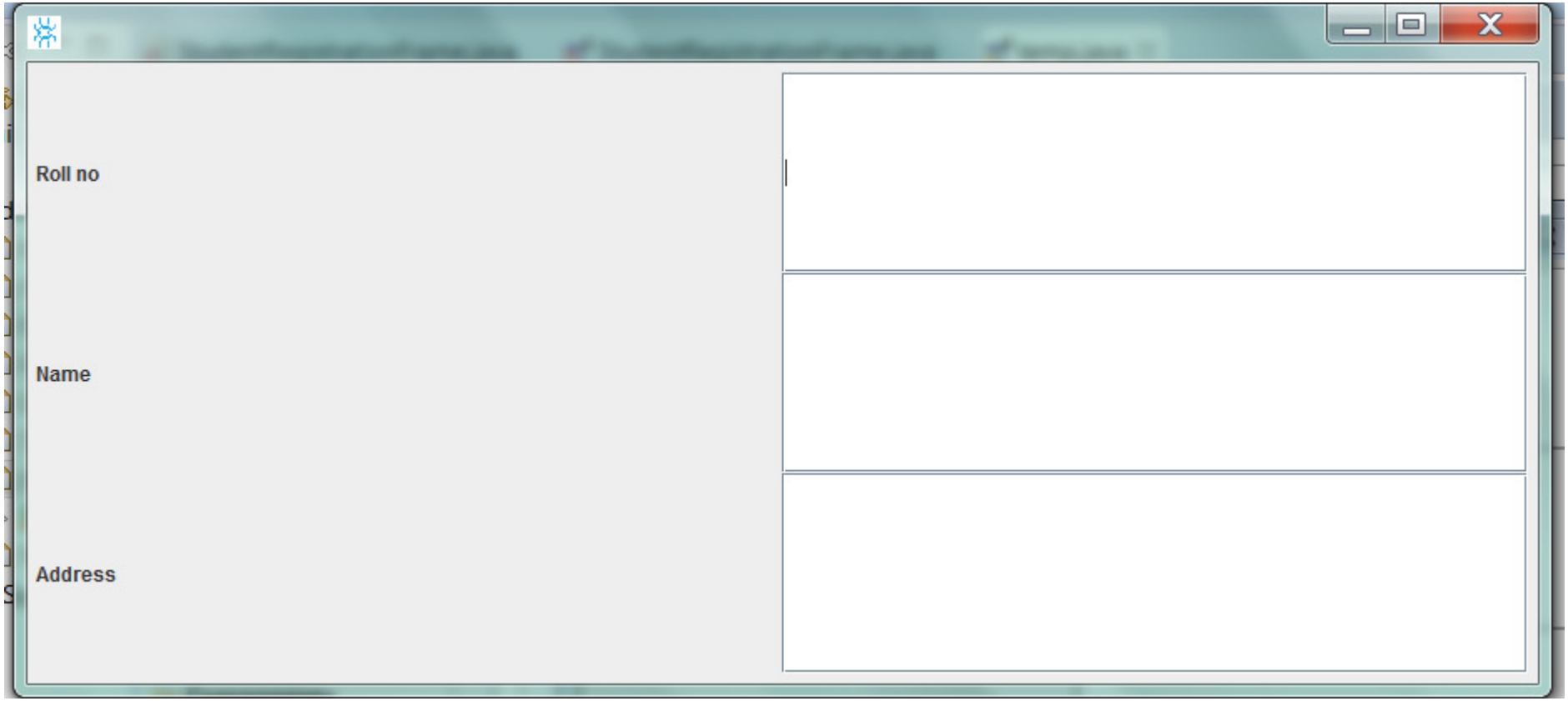
```
// retreive the current screen resolution
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
//set the frame size
frame.setSize(screenSize.width, screenSize.height);

// setBounds(100, 100, 574, 459);
// (left,top, width, height)
```

→ JFrameDemoIconImageGridLayout.java

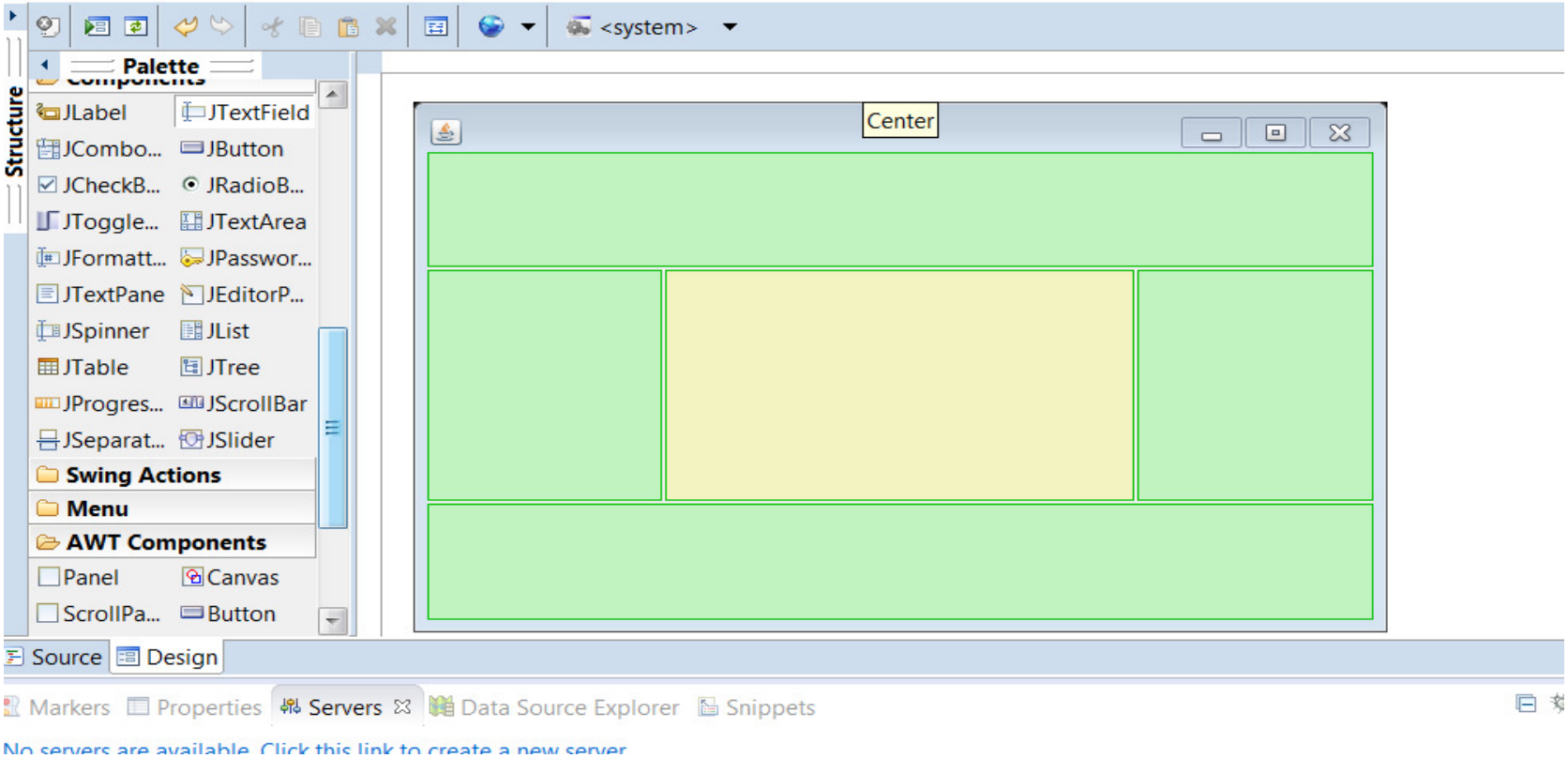
GridLayout manager

- Row and column arrangement of data



Default layout of Frame

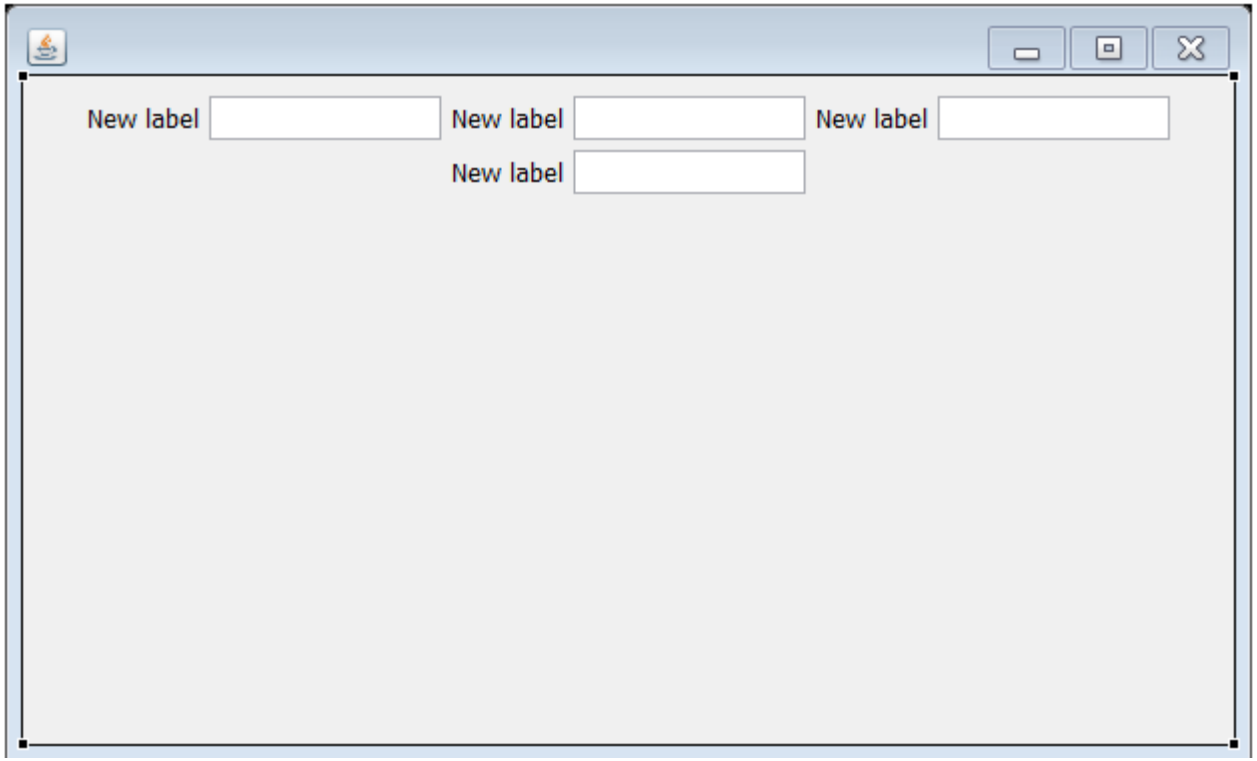
- The default layout on a JFrame is a BorderLayout.
- Add(guiControl, BorderLayout.CENTER)
- *Each of the locations of the BorderLayout can contain only one element.*



Changing the Default layout of Frame

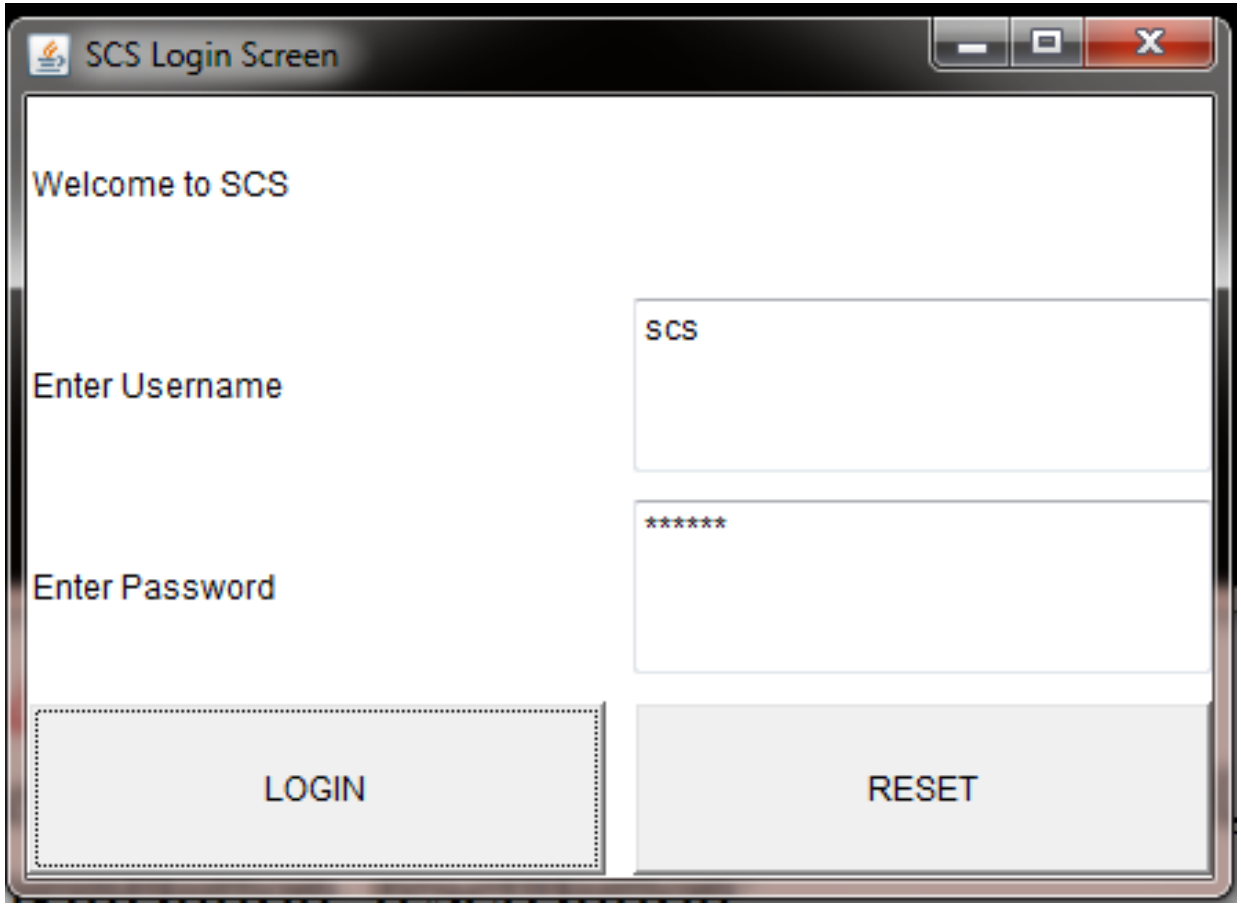
- If you want the automatic layout (automatic positing of GUI controls from left-right, top-bottom,
- you can use the FlowLayout:

  mainframe.setLayout(new FlowLayout()); //left-to-right as per frame size

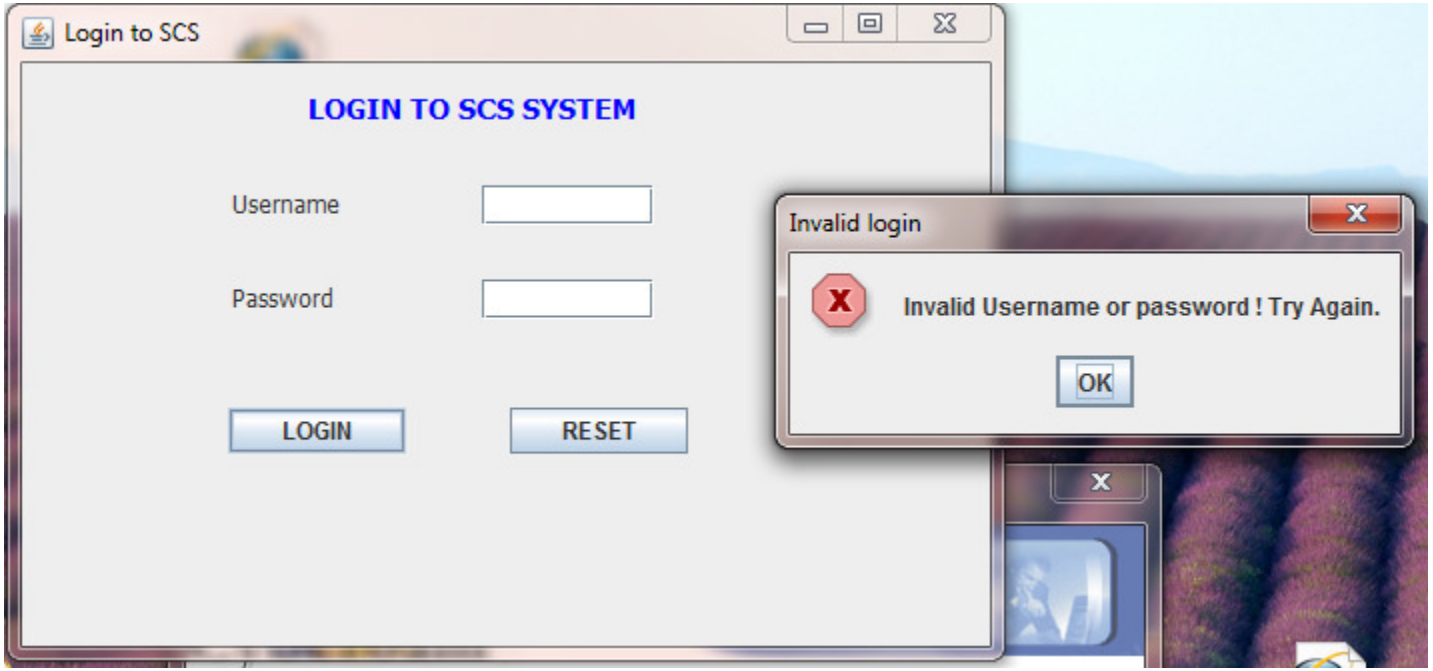


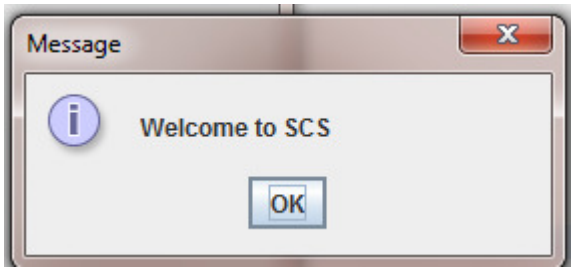
Frame Program

- → **LoginFrame.java**
- **Handcoding – GUI awt code**



Frame Program





Panel

- **Panel** is a **window** that does not contain a title bar, menu bar, icon, border"
- Other components can be added to a **Panel** –

  add() method (inherited from **Container).**
- position and resize them manually –

  setLocation(int x, int y)

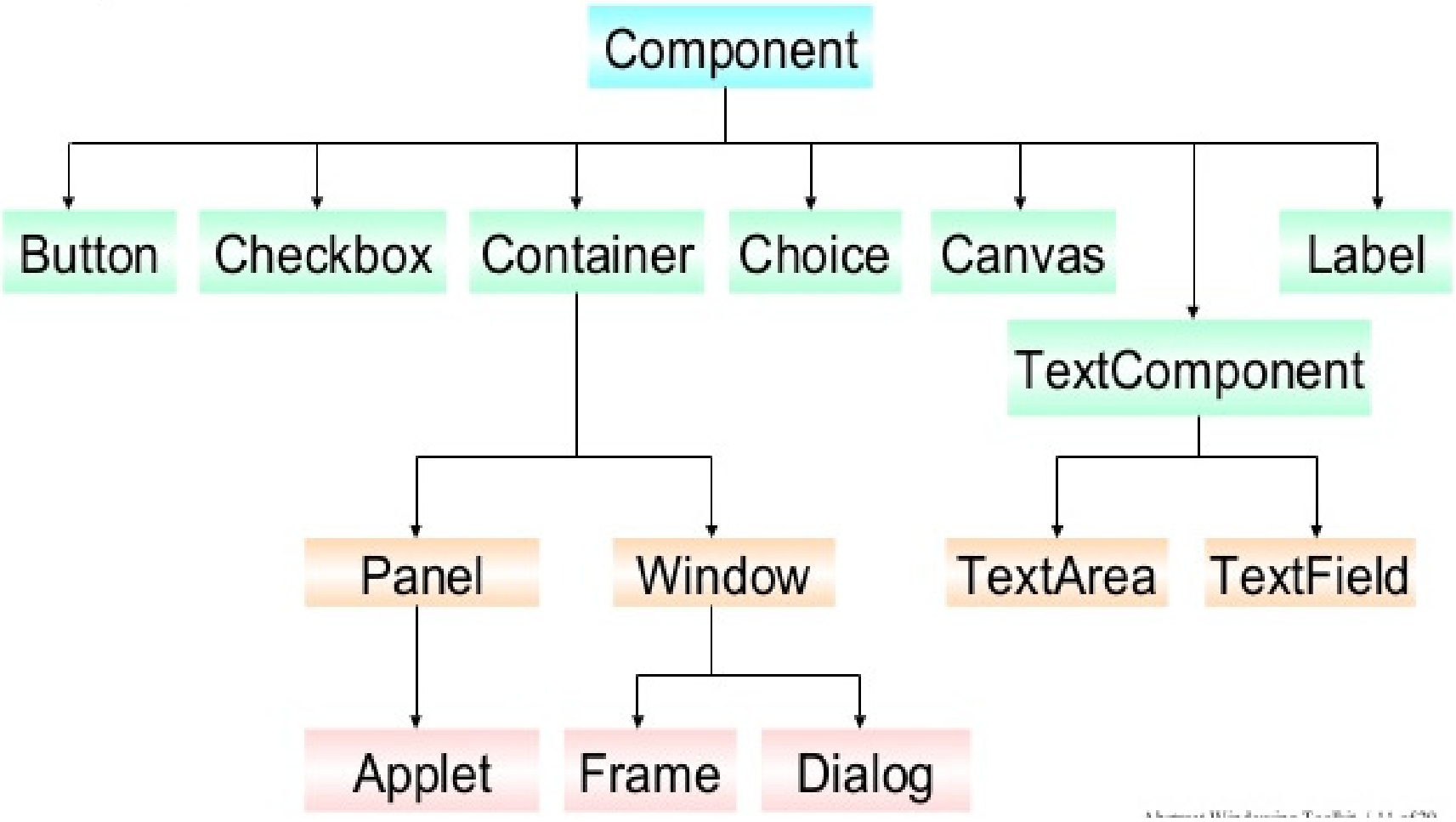
  setSize(width, height)

  defined by Component.

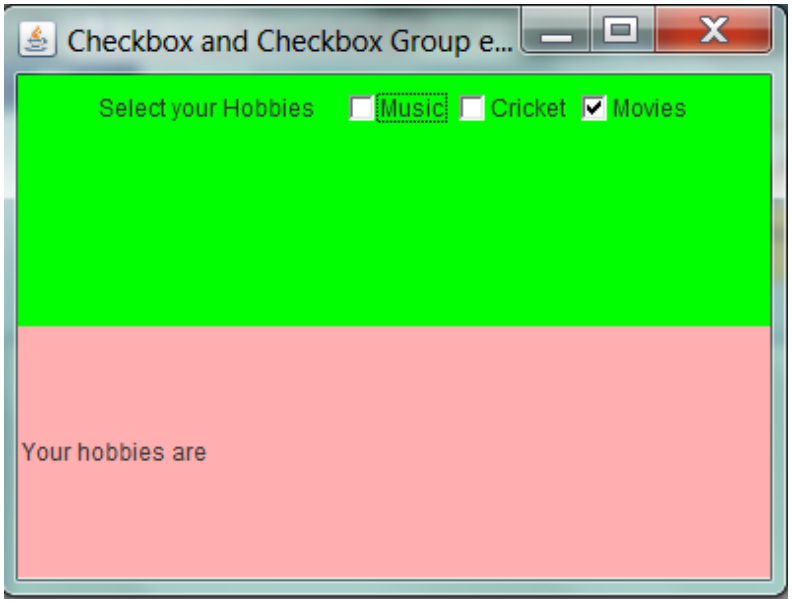
Difference - Frame v/s Panel

Sr. No.	Frame	Panel
1	Top-level container - window	Contained in some frame – to store group of controls
2	Has title bar and icon	Does not have
3	Has window resizable buttons – min, max, close	Does not have
4	Has visible, resizable borders	Does not have
5	Can have Menu bar	Does not have
6	Subclass of Window	Subclass of Container

AWT components

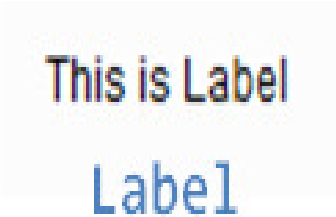


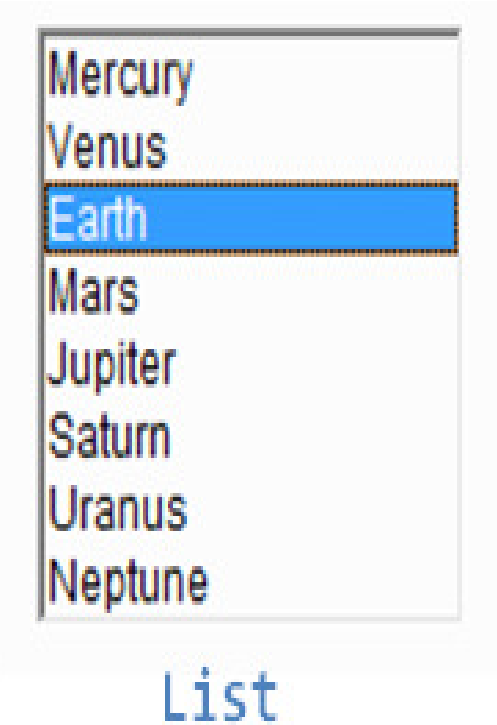
Frame with 2 panels



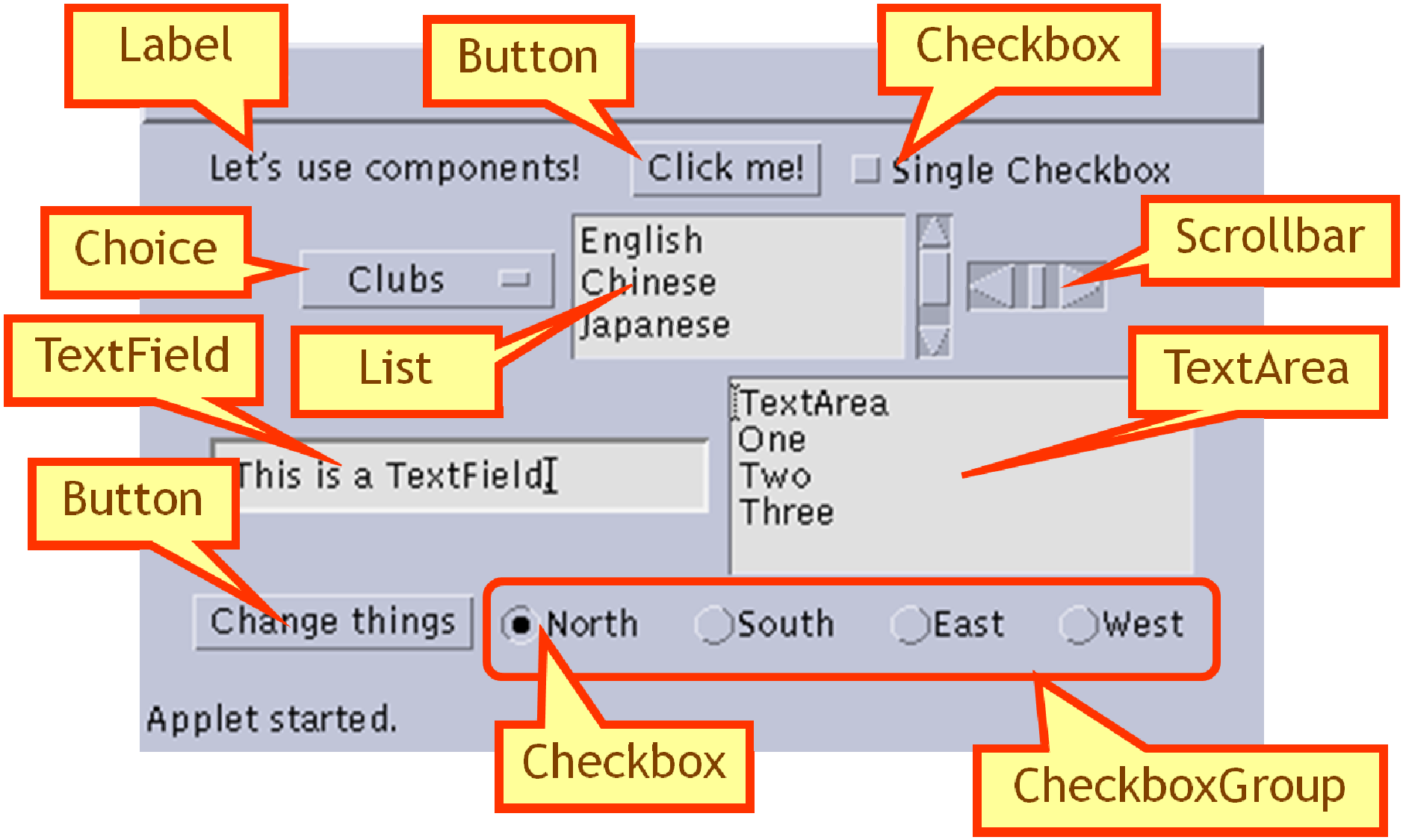
→ Refer FrameWithTwoPanels.java

AWT components





AWT components



AWT Control - Button

- Used to create labeled button
- When the button is clicked, event handling action can be taken
- Constructor
- Button b1 = new Button();
- Button(String text) - Constructs a new button with specified label.

AWT Control - Button

- Methods
- b1.setLabel("Disable middle button");
  b1.setActionCommand("disable");
- b3.setEnabled(false);
- Selecting a Button – generates event – ActionEvent
- //Listen for actions on buttons 1 and 3.
- b1.addActionListener(this);

AWT Control - Button

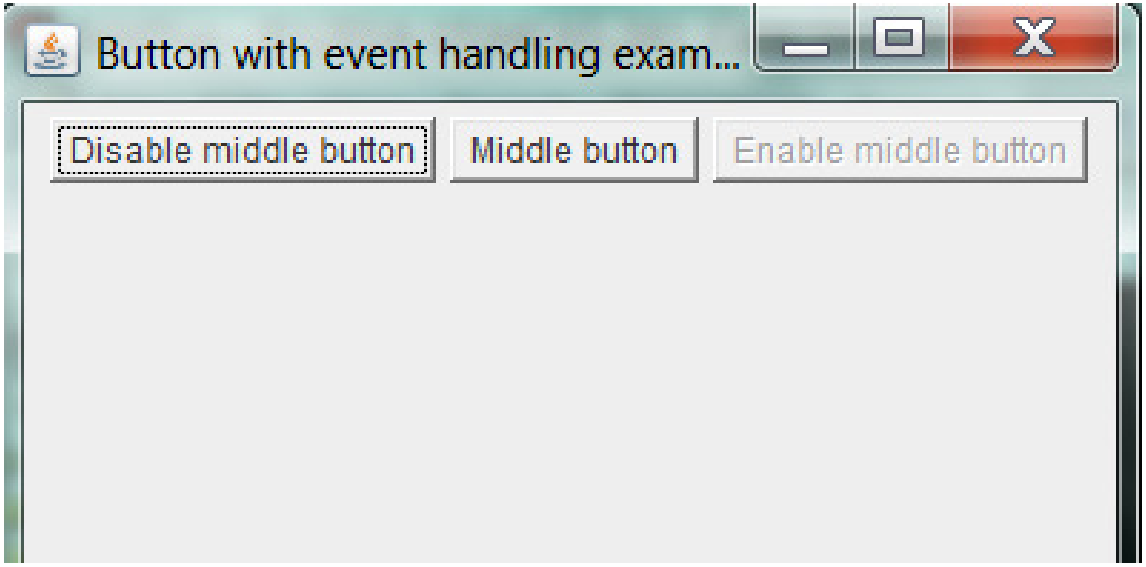
→ *Refer ButtonDemo.java*

- Button click event – ActionEvent
- To handle button event - ActionListener
-

```
public void actionPerformed(ActionEvent e) {
  String command = e.getActionCommand();
  if (command == "disable") {
        b2.setEnabled(false);
  }
}
```

AWT Control - Button

- → guicontrols\ButtonDemo.java



AWT Control - Choice

- Used to create a pop-up list (***single select combo-box)***
- Methods:

1. void add(String item) – item/text will be added
2. void insert (String item, int index) - Inserts the item into this choice at the specified position.
3. int getSelectedIndex ( ) – returns the index of the selected item
4. String getSelectedItem( ) – returns the selected item

AWT Control - Choice

- Selecting a Choice item – generates event – ItemEvent
- To handle ItemEvent – ItemListener is required.
- ChoiceDemo implements ItemListener

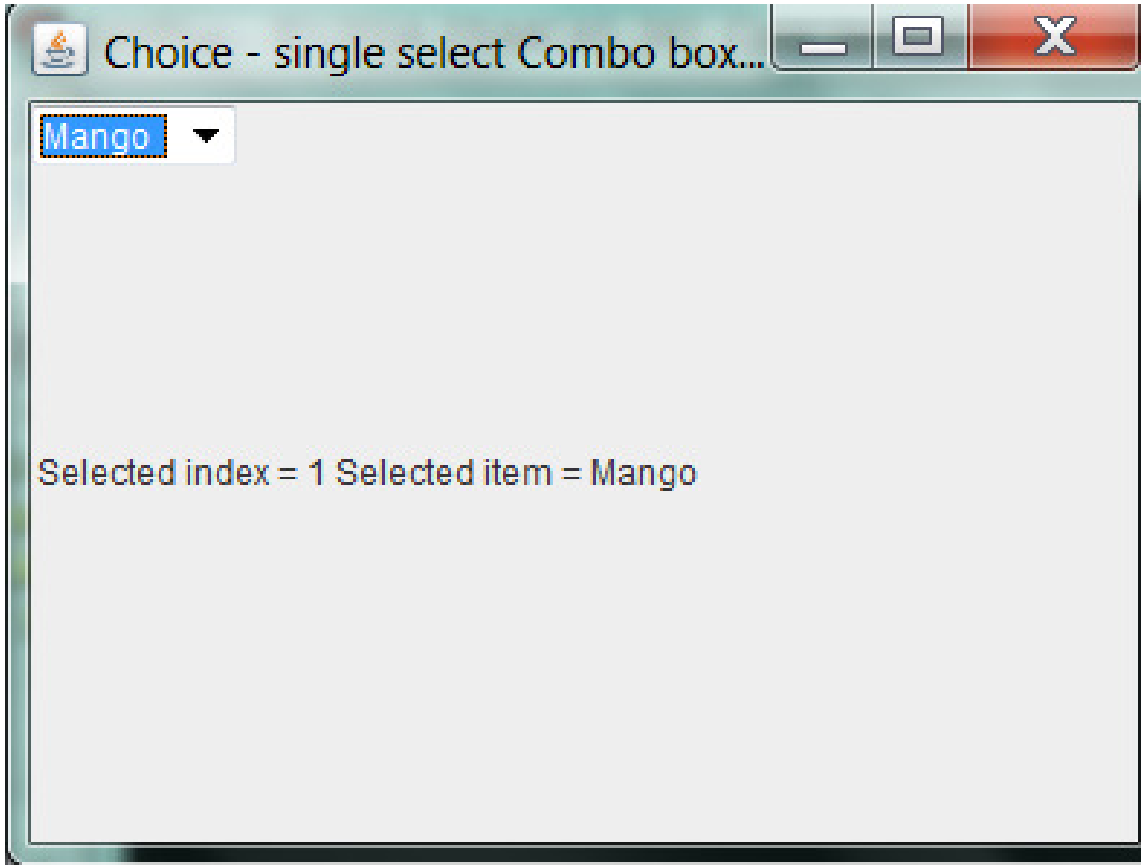
```
choice.addItemListener(this);
```

- public void itemStateChanged(ItemEvent e) {

```
    choice.getSelectedIndex();
    choice.getSelectedItem();
}
```

Choice Program

- → Refer ChoiceDemo.java



AWT Control - List

- Used to create multiple select combo-box
- Constructor

  List = new List(4); //no. of rows/visible items
- List(int rows, boolean multipleMode)

  List = new List(4, true);
- Methods:

1. list.setMultipleMode(true);
2. void add(String item) – item/text will be added
3. void add (String item, int index) - Inserts the item into this choice at the specified position.

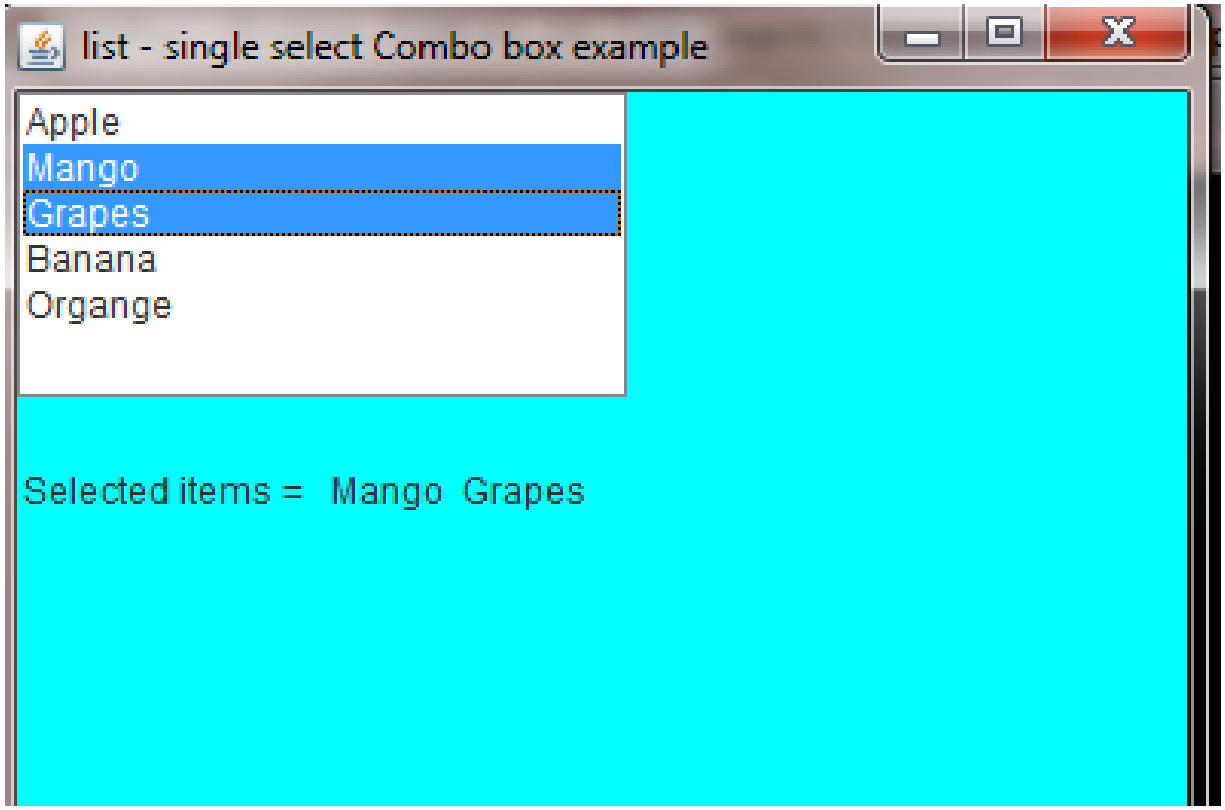
AWT Control - List

- Methods:

4. list.addItemListener(this);
5. String items[] = list.getSelectedItems();
6. String items[] = list.getItems();
7. void remove(int position)
8. void remove(String str)
9. int getItemCount() - Gets the number of items in the list.
10. int[] getSelectedIndexes() - Gets the selected indexes on the list.
11. list.setSize(200,100);

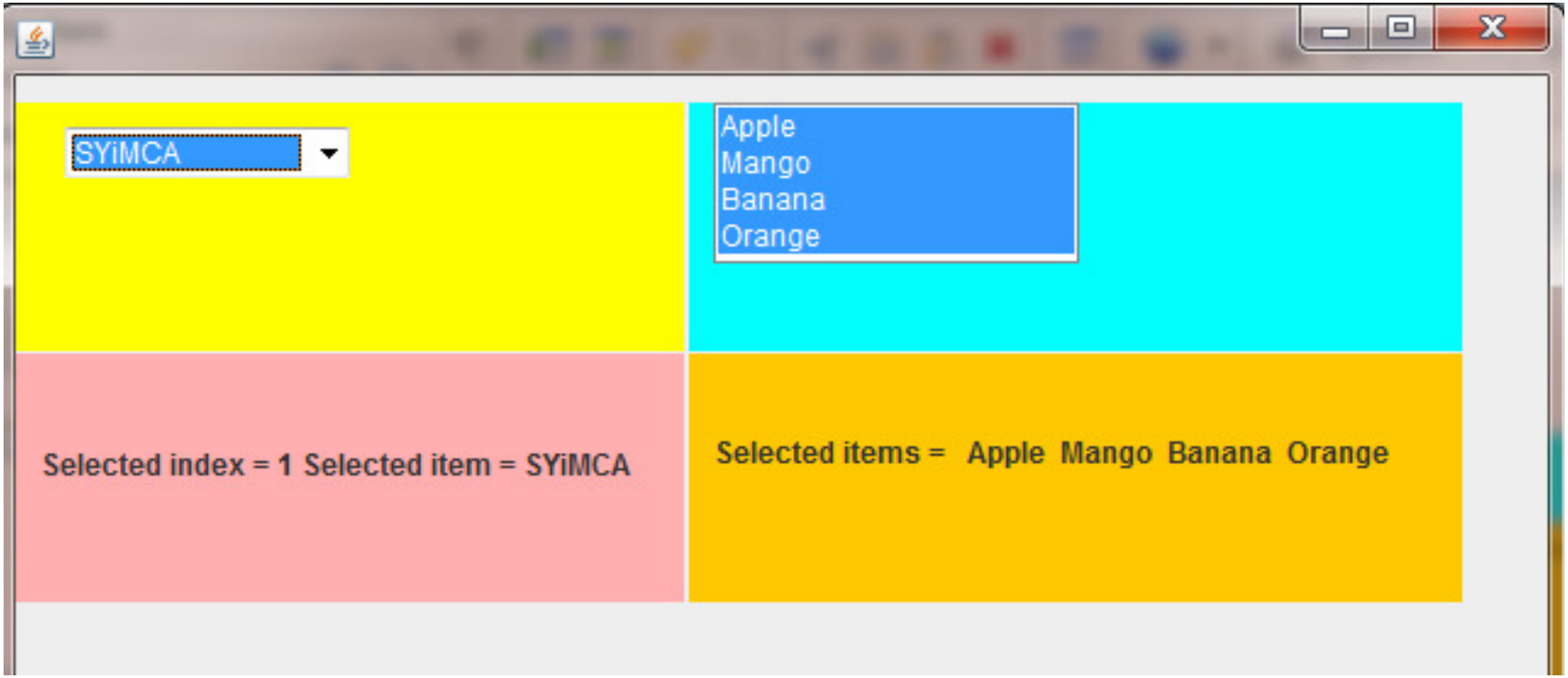
List Program

- → Refer ListMultiSelectCombboxExample.java



Choice v/s List Example

- Refer ChoiceListMultiplePanels.java



AWT Control – Checkbox

- checkbox – java.awt.Checkbox
- It can be "on" (true) or "off" (false) state.

Methods of Checkbox

- cb1 = new Checkbox();
  cb1.setLabel("Checkbox 1");
- cb2 = new Checkbox("Checkbox 2");
- cb3 = new Checkbox("Checkbox 3");

Checkbox Methods

- To check a checkbox

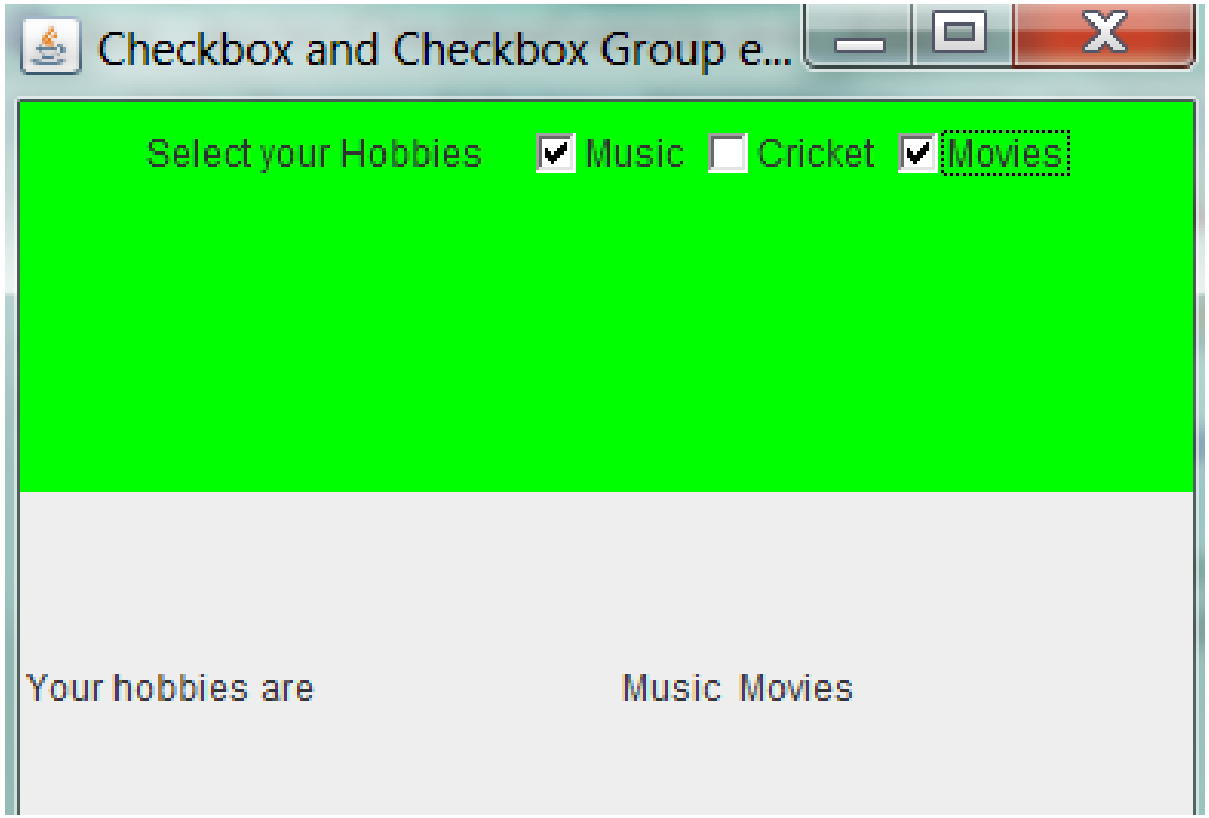
  cb3.setState(true);

- To obtain status of checkbox

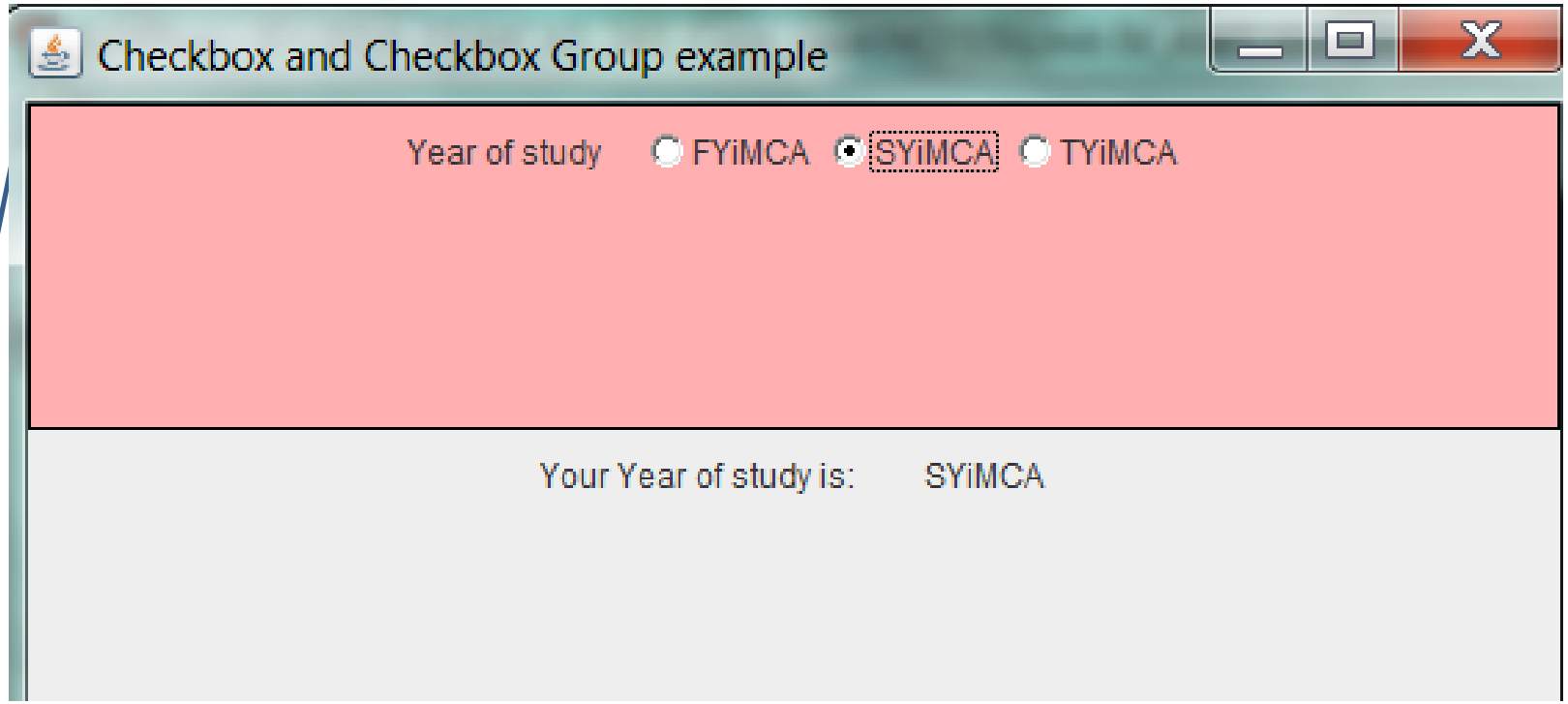
  boolean isChecked = cb3.getState();

Checkbox Program

- Refer CheckboxExample.java



AWT Control –RadioButton



-CheckboxGroup – used to create RadioButton

-only one Checkbox can be selected at a time

AWT Control – CheckboxGroup

- Used to create radio button
- Constructor

  checkboxGroup = new CheckboxGroup( );
- cb1 = new Checkbox("FY", checkboxGroup, false);
- cb2 = new Checkbox("SY", checkboxGroup, false);
- Methods:

1. Checkbox getSelectedCheckbox() - Gets the current choice from this check box group.
2. setSelectedCheckbox(Checkbox box) - Sets the specified check box as selected

AWT Control – CheckboxGroup

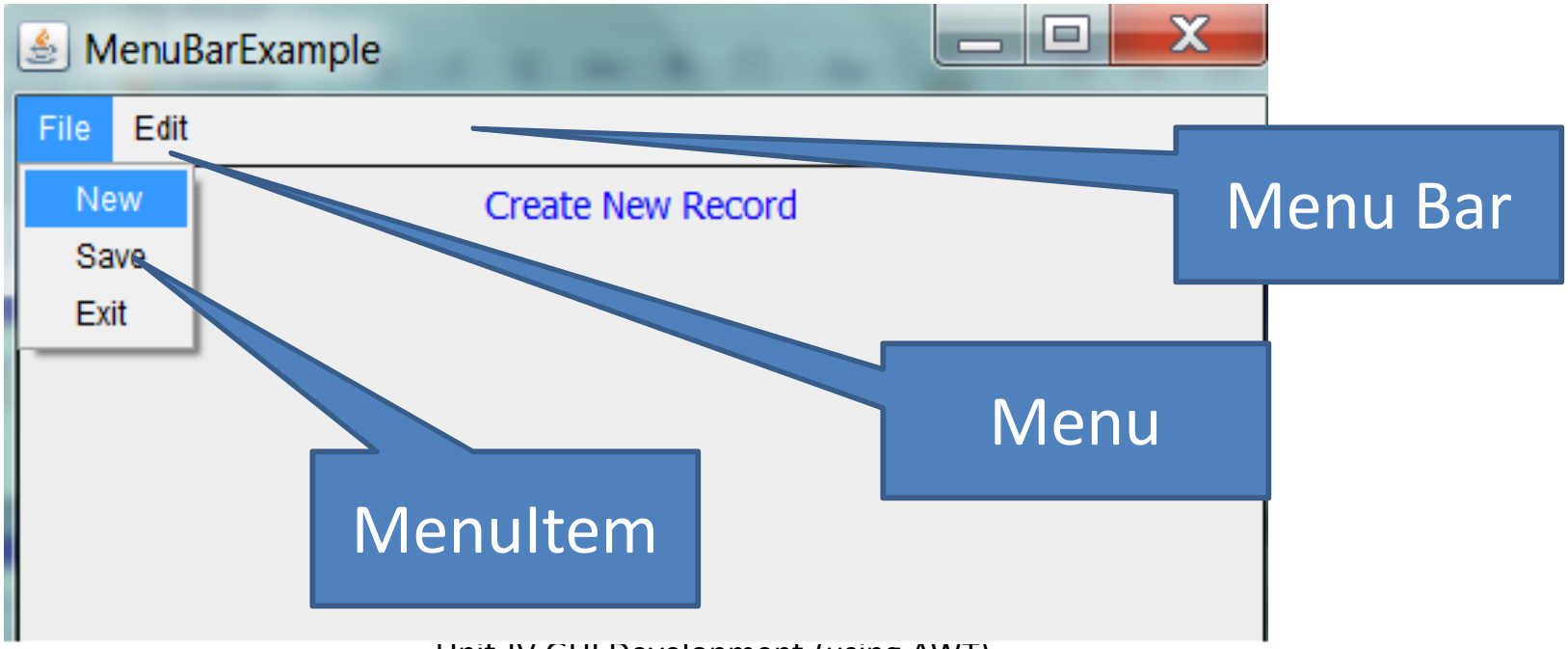
- Event Handling Code:
- class implements implements Itemlistener {

```
public void itemStateChanged(ItemEvent e) {
    String year="";
    if(cb1.getState())
            year=" FYiMCA ";
    else if (cb2.getState())
            year = " SYiMCA";

    label.setText(year);
}
```

AWT Control – MenuBar

- Provides menu bar bound to a frame
- Multiple menus and menu-items can be added
- Menu Container Hierarchy
  - MenuBar contains many Menus
  - 1 Menu can contain many MenuItems



AWT Control – MenuBar

- import java.awt.*;
- MenuBar menubar = new MenuBar();
- //Create main menus

  Menu fileMenu = new Menu("File");
- //create Menu-items in File Menu

  MenuItem newMenuItem = new MenuItem("New");

  MenuItem saveMenuItem = new MenuItem("Save");
- // Attach menu-items to main-menus

  fileMenu.add(newMenuItem);

  fileMenu.add(saveMenuItem);

AWT Control – MenuBar

- //Attach main-menus to MenuBar

  menuBar.add(fileMenu);

- //add Menubar to the frame

  frameObject.setMenuBar(menuBar);

MenuBar – Event Handling

- Clicking a menu item – generates – ActionEvent
- So, to handle menu click event - ActionListener is required to be created
- public class MenuBarExample extends JFrame

  implements ActionListener {
- Event handling method – to be define in same class

```
public void actionPerformed(ActionEvent e) {
   if(e.getActionCommand().equals("New")) {
            getContentPane().removeAll();
            getContentPane().add(new WelcomeFrame());
```

MenuBar – ActionListener

- Register ActionListener with MenuItem

  newMenuItem.setActionCommand(“New");

  newMenuItem.addActionListener(this);

- When the user will Click the New MenuItem – ActionEvent is generated (at runtime).
- JRE will notify registered ActionListener – and event handling method – actionPerformed - will be called

How to Switch Frames in MenuBar

- How to close one window and open Second Frame
- To close the current window (Frame)

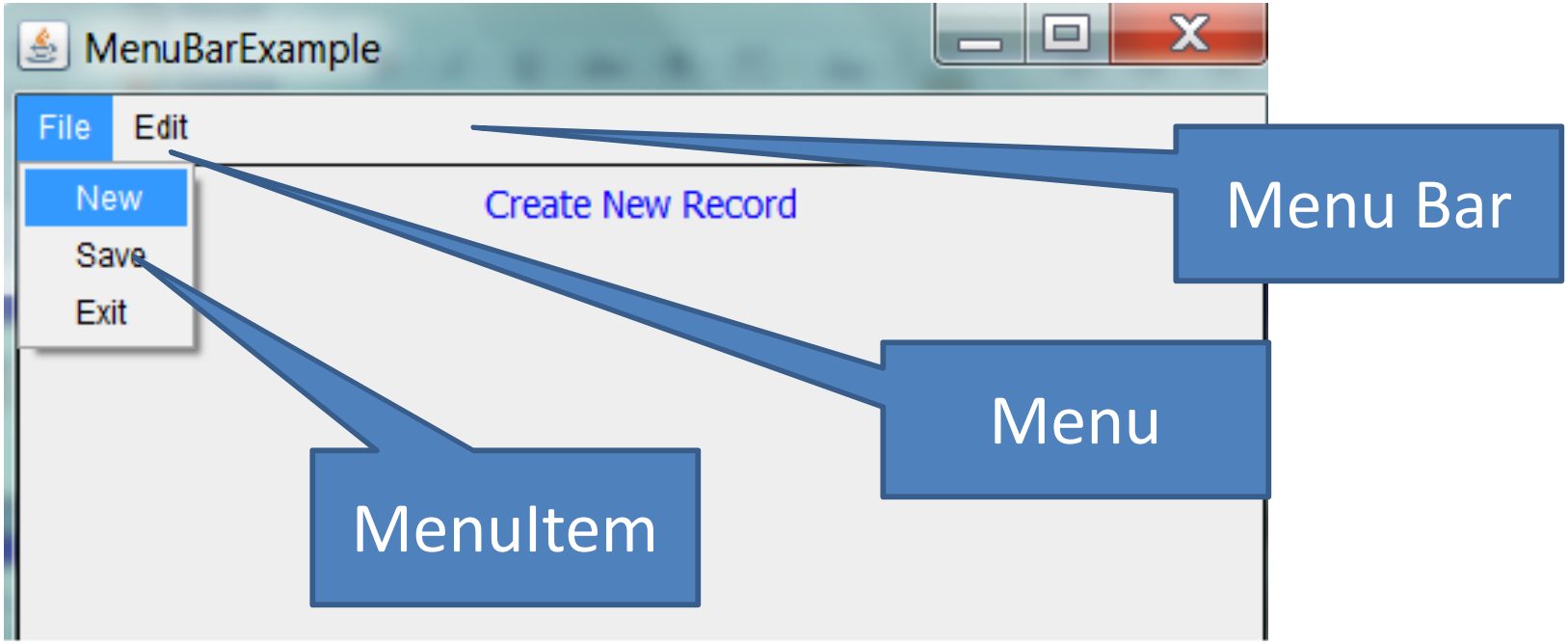
```
frame1.setVisible(false);
frame1.dispose( );      //release resources
```

- Create object of second Frame

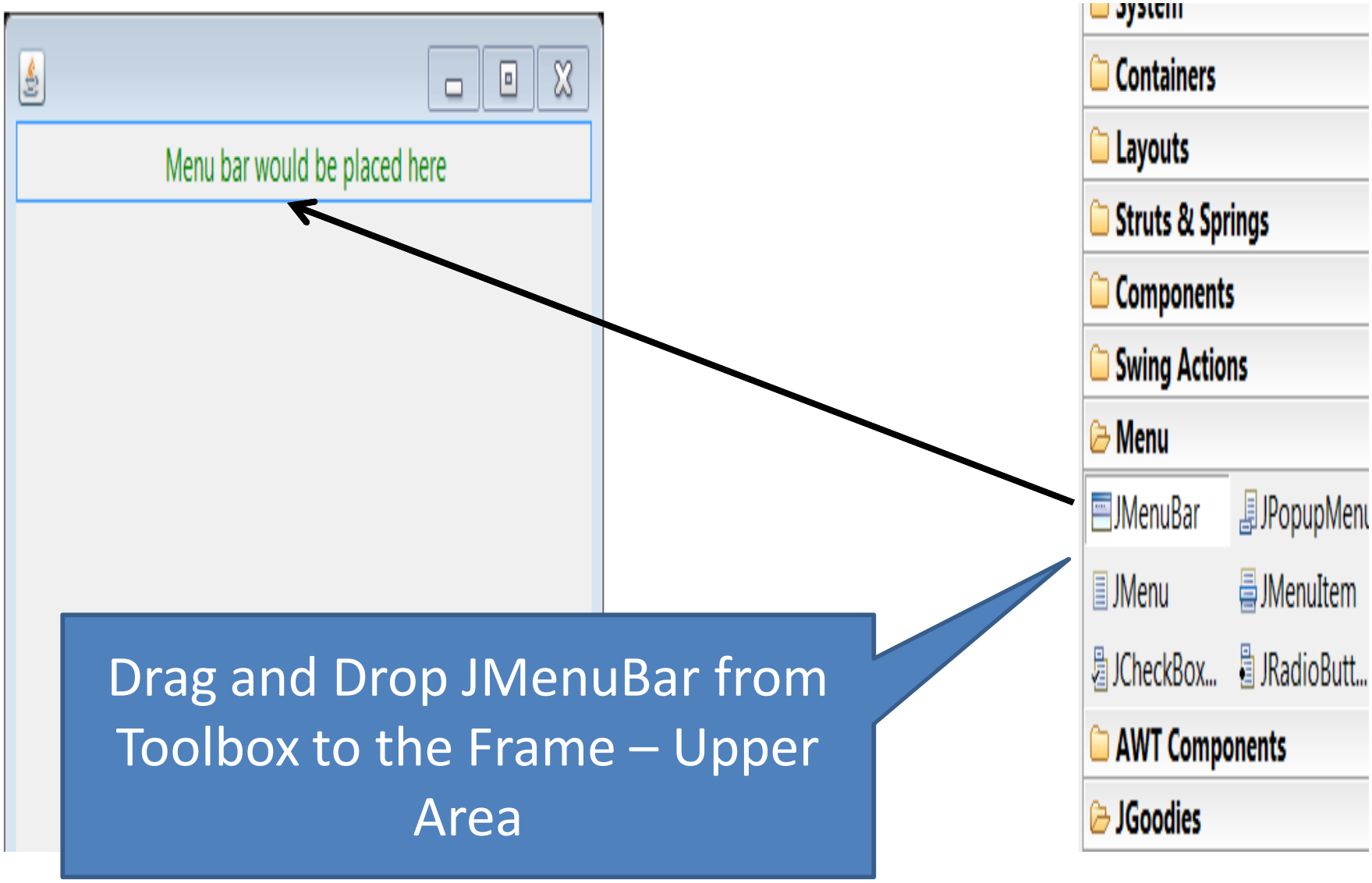
```
SecondFrame frame2 = new SecondFrame();
frame2.setVisible(true);
```

MenuBar Example Program

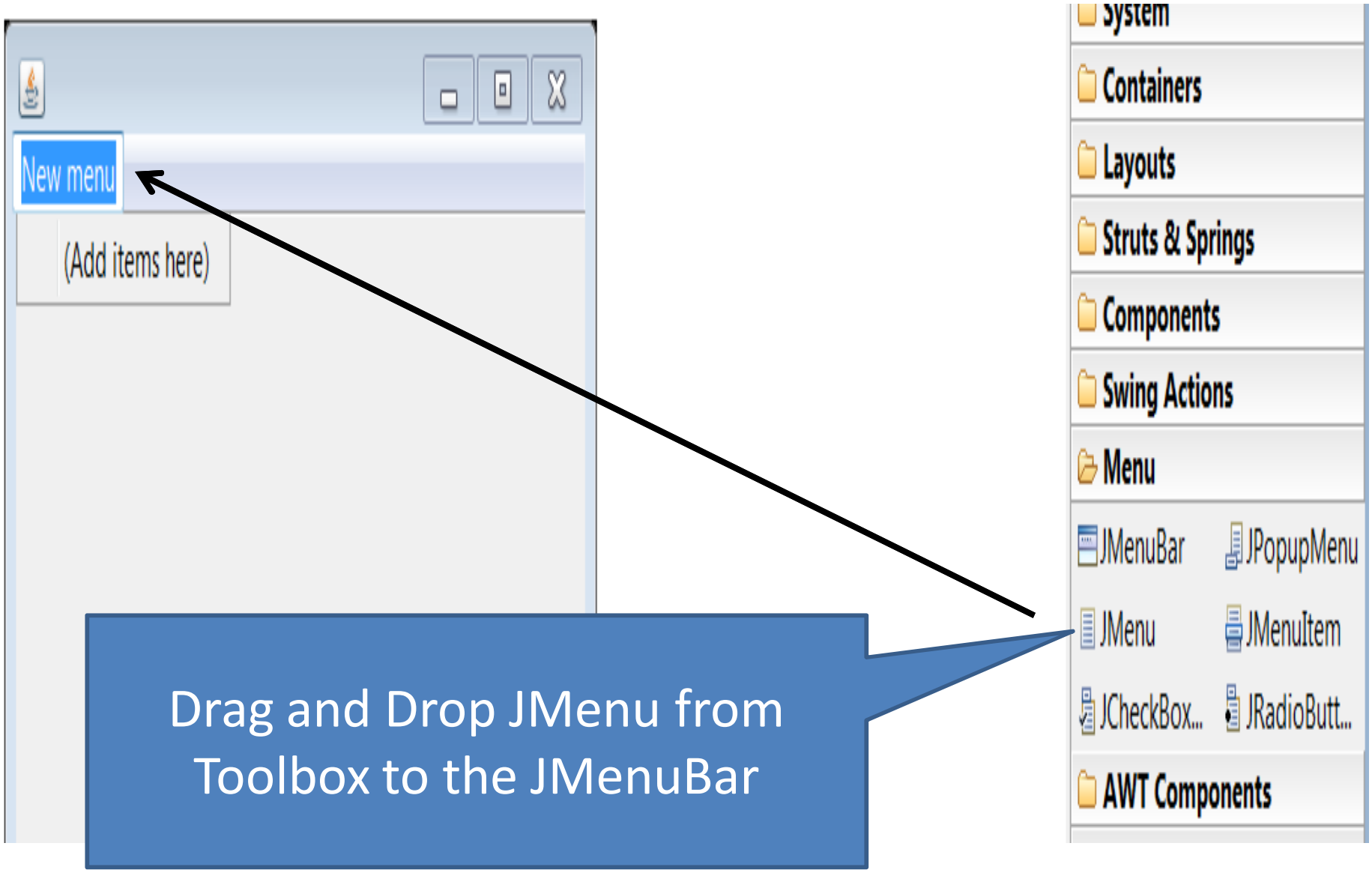
- Refer → MenuBarExample.java



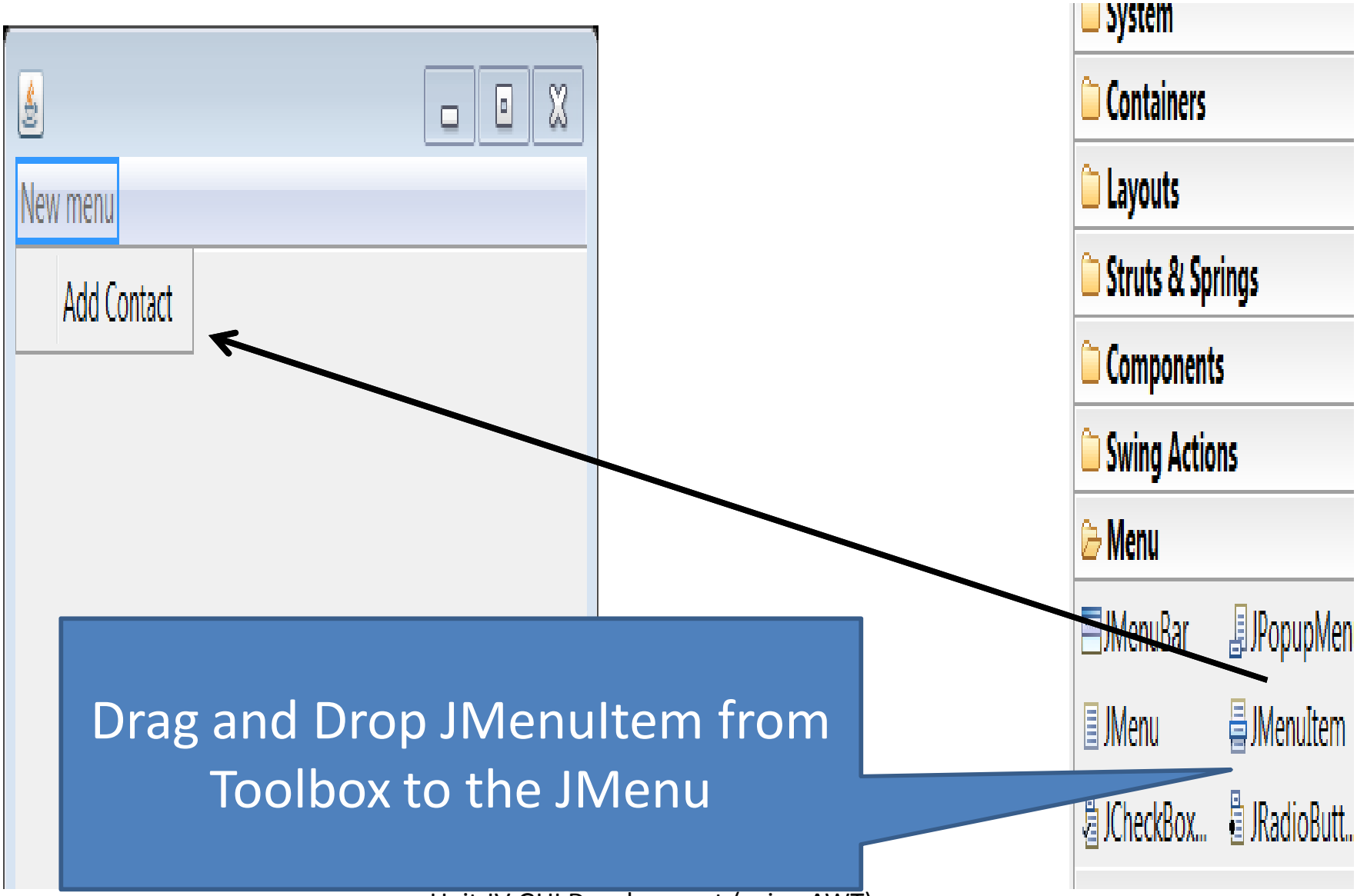
MenuBar – Using Eclipse - WindowBuilder



MenuBar – Step2- Drag JMenu



MenuBar – Step3- Drag JMenuItem



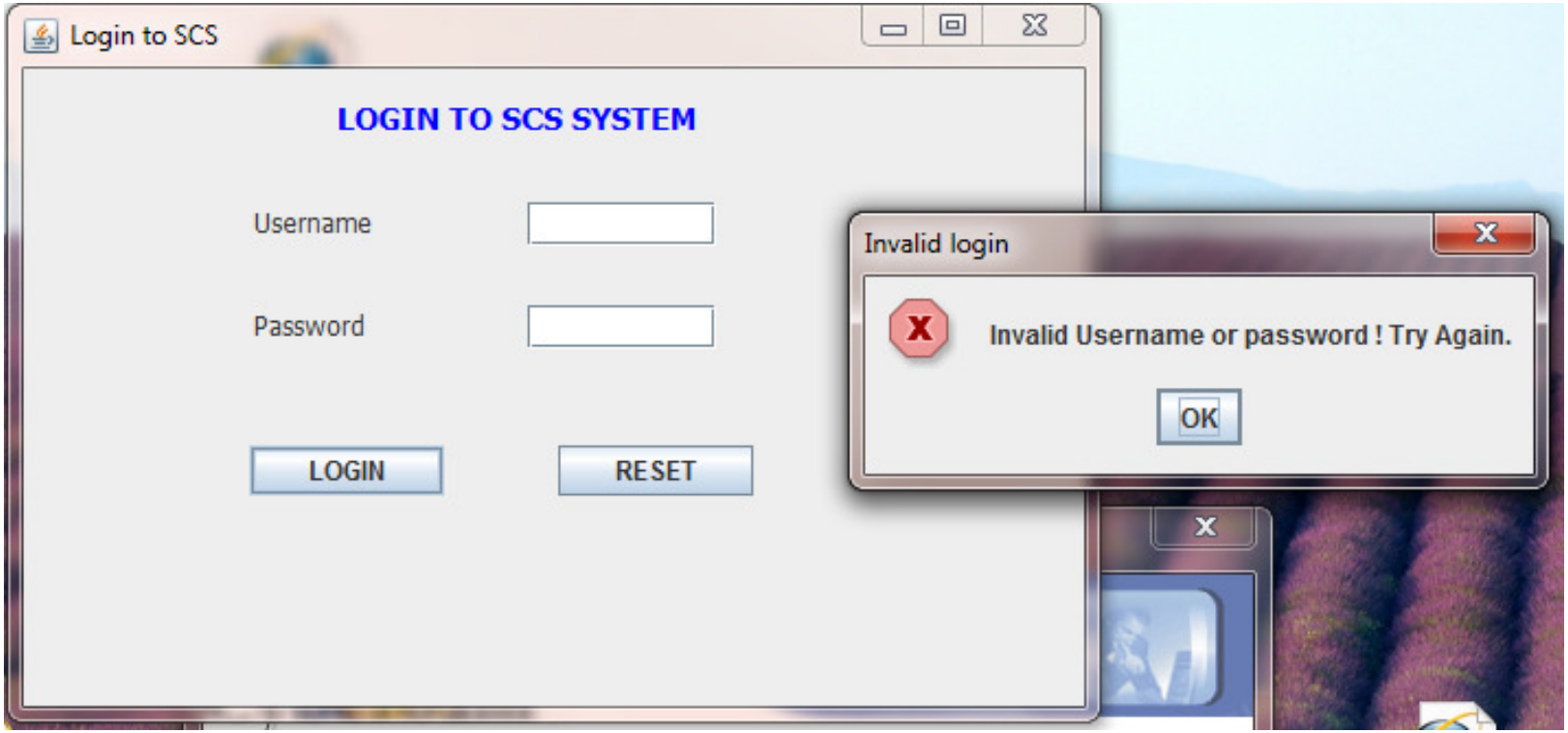
Dialog Boxes (JOptionPane)

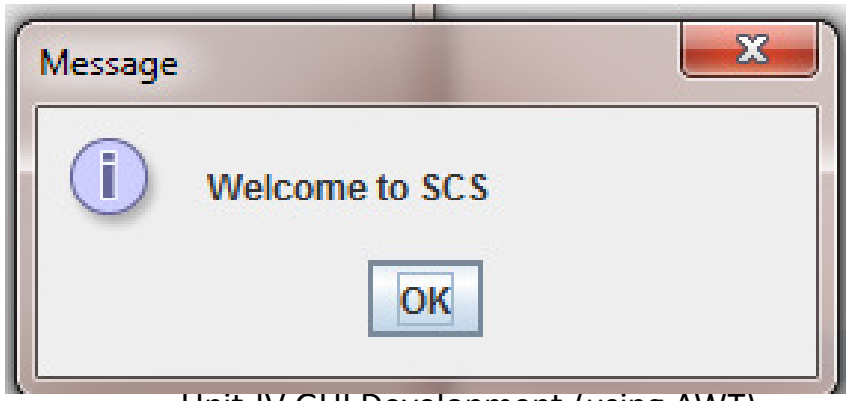
http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html#features

```
if(username.equals(uname) & password.equals(pass)) {
        JOptionPane.showMessageDialog(this, "Welcome to SCS");

} else {
        JOptionPane.showMessageDialog(this, "Invalid Username or
        password ! Try Again.", "Invalid login",
        JOptionPane.ERROR_MESSAGE);
//frameObject, error message, title of dialog box,type of dialog box
}
```

Dialog Boxes (JOptionPane)



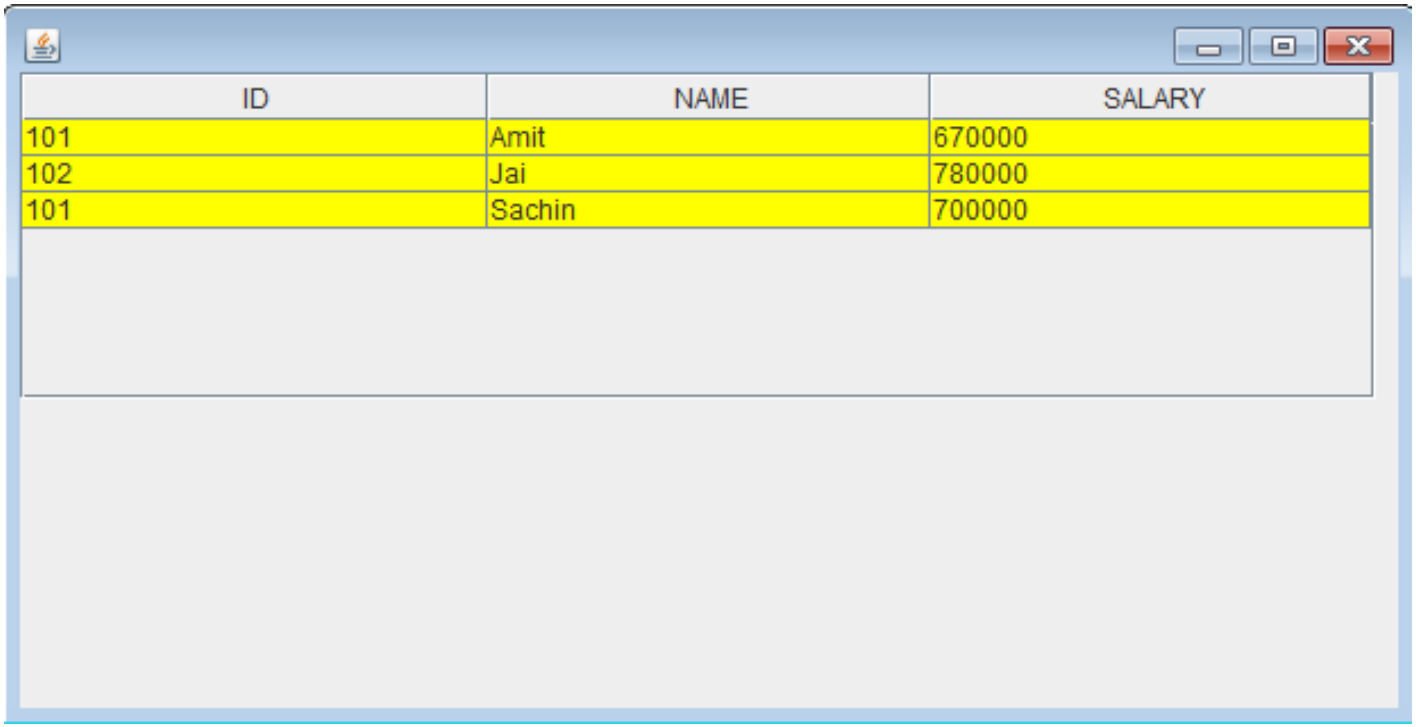


JTable

- used to display the data on row and column format.
- Constructors of JTable class:

  JTable(): creates a table with empty cells.

  JTable(Object[][] data, Object[] columns): creates a table with the specified data.



ID	NAME	SALARY
101	Amit	670000
102	Jai	780000
101	Sachin	700000

Jtable (Refer JTableDemo.java)

```
private JTable table;
public JTableDemo() {
    String columns[]={"ID","NAME","SALARY"};
    String data[][]={
    {"101","Amit","670000"},
    {"102","Jai","780000"},
    {"101","Sachin","700000"}};
    table = new JTable(data,columns);
    JScrollPane sp=new JScrollPane(table);
    frameObject.add(sp);
}
```

How to add a image in JFrame

- used to display the data on row and column format.
- JLabel label= new JLabel("New label");
- label.setIcon(new ImageIcon("C:/Pictures/Icon.jpg"));
- Label.setSize(400,300); //size of image
- frame.add(label);

Difference - AWT v/s Swing

Sr. No.	AWT	Swing
1	Look and feel is OS based (use native code)	GUI is OS independent (pure java based)
2	Heavy weight (memory)	Light weight and efficient
3	Limited functionality	Additional features/capability – drag & drop
4	Basic GUI controls, with basic features	New controls – JTable and Jtree, JPasswordField

JFrame

- Represents a top-level window
- JFrame is the subclass of Frame class
- When a JFrame window is created, its size is (0,0) and is invisible
- Inherits methods from Classes: java.awt.Container, java.awt.Component, java.lang.Object
- Constructors:
- JFrame(String title)
- Creates an invisible window without a title given by the String title

JFrame

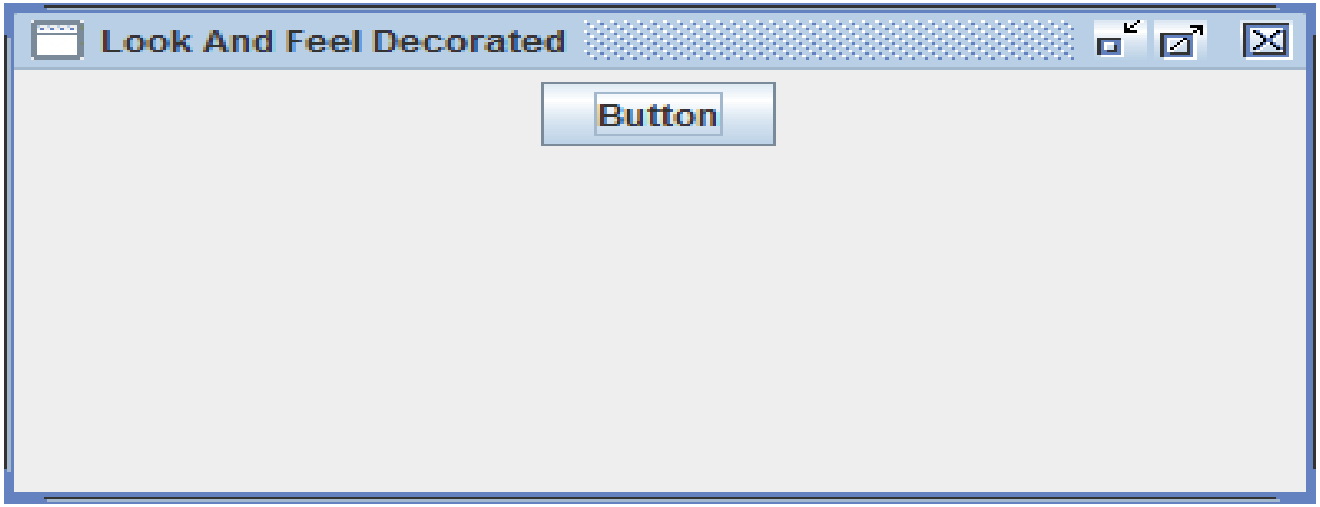
- Methods:
- String getTitle() - Returns a string representing the title of the frame
- void setTitle(String Title) - Sets the title of the frame to this string
- void setVisible(booleanb) - Shows or hides this frame window
- void setSize(intwidth, intheight) - Sets the size of the window to the specified width and height in pixels
- void setIconImage(Image image) - Sets the image to be displayed as the icon for this window.

Jframe methods (contd.)

- Methods:
- When you want to close the current window and open a new window
- public void dispose() – release all the resouces & memory occupied by current window

JFrame

- Methods:
- static void setDefaultLookAndFeelDecorated(boolean defaultLookAndFeelDecorated)
- By Default- Jframe has Native OS specific – L & F
- By setting it to true, you will get Java/OS independent Look and Feel (called Metal Look and Feel)



Jframe - contentpane

- Jframe contains a Jpanel – called as contentpane
- As a rule– all GUI controls (except menubar) are added in this content pane.
- content pane is object of Jpanel
- Container getContentPane( ) - Returns the contentPane object for this frame.
- void setContentPane(Container contentPane) - sets the contentPane property.