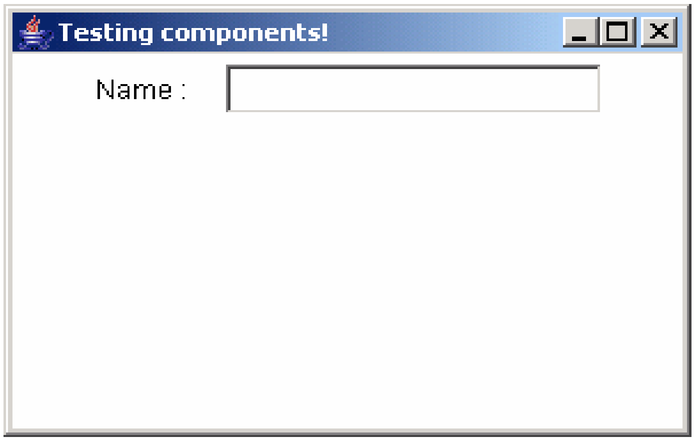
1. Type the below code to create an AWT window. Save the file as AcceptNameAWT.java

```
import java.awt.*;
class AcceptNameAWT extends Frame
{
   TextField txtName = new TextField(20);
   Label lblName = new Label("Name :");
   public AcceptNameAWT (String title)
   {
      super(title);
      setLayout(new FlowLayout());
      add(lblName);
      add(txtName);
   }
   public static void main(String args[])
   {
      AcceptNameAWT  ObjAccName  =  new  AcceptNameAWT  ("Testing
components!");
      ObjAccName.setSize(300,200);
      ObjAccName.show();
   }
}
```

The output of the program:



- Change the program code from AWT into Swing. Save the file name as AcceptNameSwing

```
import java.awt.*;
import javax.swing.*;
class AcceptNameSwing extends JFrame
{
   JTextField txtName = new JTextField(20);
   JLabel lblName = new JLabel("Name :");
   public AcceptNameSwing (String title)
   {
      super(title);
      setLayout(new FlowLayout());
      getContentPane().add(lblName);
      getContentPane().add(txtName);
   }
   public static void main(String args[])
   {
      AcceptNameSwing ObjAccName = new AcceptNameSwing ("Testing
components!");
      ObjAccName.setSize(300,200);
      ObjAccName.show();
   }
}
```

You should compare the code and result of two above programs.

Ø Change the program code from application into applet. Save the file name as AcceptNameApplet.java

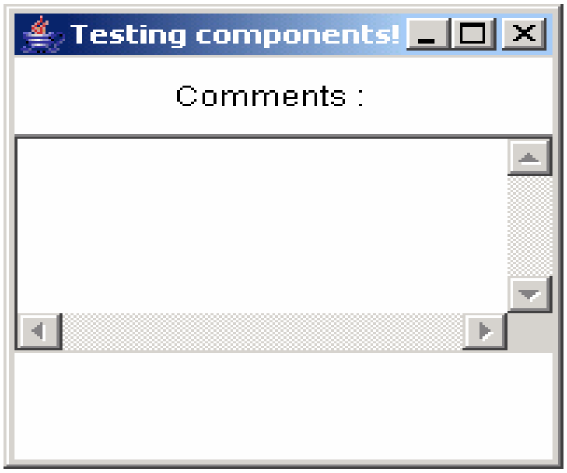
```
import java.awt.*;
import javax.swing.*;
public class AcceptNameApplet extends JApplet
{
   JTextField txtName = new JTextField(20);
   JLabel lblName = new JLabel("Name :");
   public void init()
   {
      setLayout(new FlowLayout());
      getContentPane().add(lblName);
      getContentPane().add(txtName);
   }
}
```

You should compare the code and result of three above programs.

2. Type the following code

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class TextComments extends JFrame
{
   JTextArea txtComment = new JTextArea(5,25);
   JLabel lblCom = new JLabel("Comments :");
   public TextComments(String title)
   {
      super(title);
      setLayout(new FlowLayout());
      add(lblCom);
      add(txtComment);
      addWindowListener(new WindowAdapter()
      {
         public void windowClosing(WindowEvent we)
         {
            setVisible(false);
            System.exit(0);
         }
      });
   }

   public static void main(String args[])
   {
      TextComments   ObjComment   =   new   TextComments("Testing
components!");
      ObjComment.setSize(200,200);
      ObjComment.show();
   }
}
```

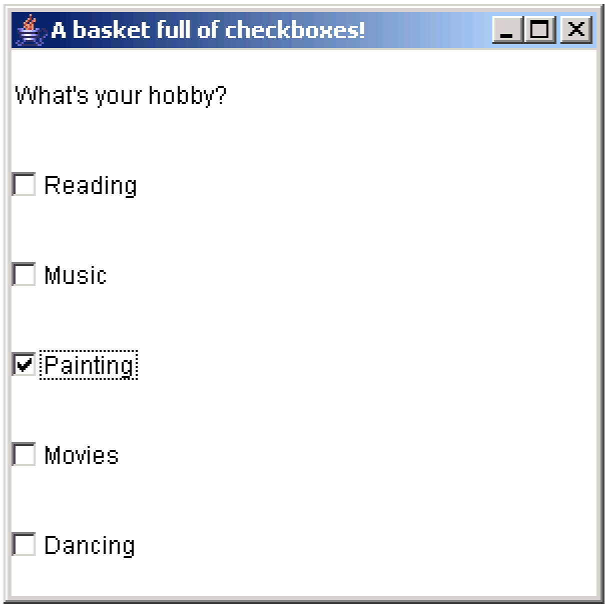
The result:



3. Type the below code snippet:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class Hobbies extends JFrame
{
   JCheckBox cboxRead = new JCheckBox("Reading",false);
   JCheckBox cboxMus = new JCheckBox("Music",false);
   JCheckBox cboxPaint = new JCheckBox("Painting",false);
   JCheckBox cboxMovie = new JCheckBox("Movies",false);
   JCheckBox cboxDance = new JCheckBox("Dancing",false);
   JLabel lblQts = new JLabel("What's your hobby?" );
   public Hobbies(String str )
   {
      super(str);
      setLayout(new GridLayout(6,1));
      Container cp = getContentPane();
      cp.add(lblQts);
      cp.add(cboxMus);
      cp.add(cboxRead);
      cp.add(cboxPaint);
      cp.add(cboxMovie);
      cp.add(cboxDance);
      addWindowListener(new WindowAdapter()
      {
         public void windowClosing(WindowEvent we)
         {
            setVisible(false);
            System.exit(0);
         }
      });
   }

   public static void main(String args[])
   {
      Hobbies ObjHobby = new Hobbies ("A basket full of
checkboxes!");
      ObjHobby.setSize(300,300);
      // Objhobby.pack();
      ObjHobby.show();
   }
}
```

The result:



A basket full of checkboxes!

What's your hobby?

☐ Reading

☐ Music

☑ Painting

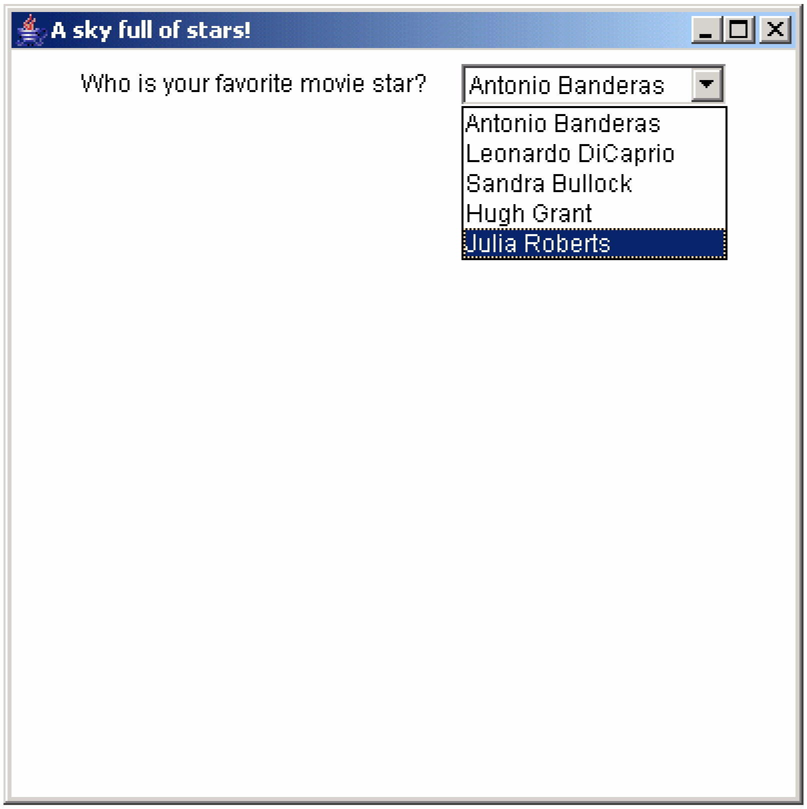
☐ Movies

☐ Dancing

4. Type the following code snippet:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class Stars extends JFrame
{
   JComboBox moviestars = new JComboBox();
   JLabel lblQts = new JLabel("Who is your favorite movie
star?");
   public Stars(String str)
   {
      super(str);
      setLayout(new FlowLayout());
      moviestars.addItem("Antonio Banderas");
      moviestars.addItem("Leonardo DiCaprio");
      moviestars.addItem("Sandra Bullock");
      moviestars.addItem("Hugh Grant");
      moviestars.addItem("Julia Roberts");
      getContentPane().add(lblQts);
      getContentPane().add(moviestars);
      addWindowListener(new WindowAdapter()
      {
         public void windowClosing(WindowEvent we)
         {
            setVisible(false);
            System.exit(0);
         }
      });
   }
   public static void main(String args[])
   {
      Stars ObjStar = new Stars ("A sky full of stars!");
      ObjStar.setSize(400,400);
      ObjStar.show();
   }
}
```

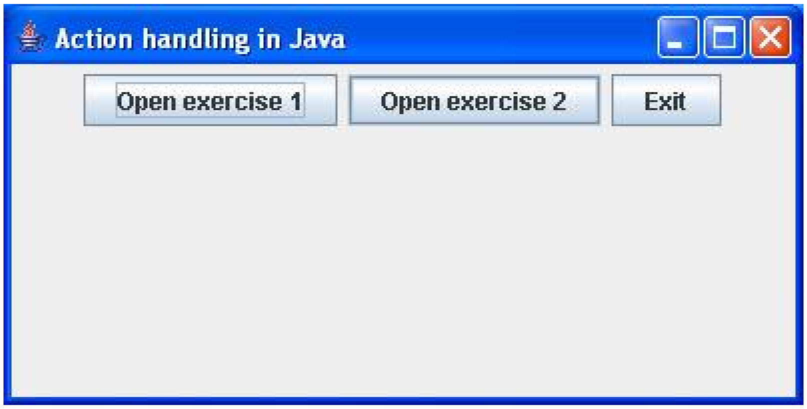
The result:



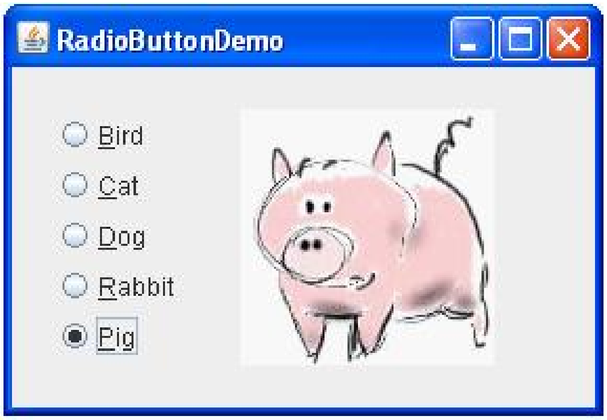
5. Type the below code:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class ActionHandler extends JFrame implements ActionListener
{
   JButton bEx1 = new JButton("Open exercise 1");
   JButton bEx2 = new JButton("Open exercise 2");
   JButton bExit = new JButton("Exit");
   public ActionHandler(String str)
   {
      super(str);
      Container cp = getContentPane();
      setLayout(new FlowLayout());
      bEx1.addActionListener(this);
      cp.add(bEx1);
      bEx2.addActionListener(this);
      cp.add(bEx2);
      bExit.addActionListener(this);
      cp.add(bExit);
      // Exit the application if user clicks close button
      addWindowListener(new WindowAdapter()
      {
         public void windowClosing(WindowEvent we)
         {
            setVisible(false);
            System.exit(0);
         }
      });
   }
   public void actionPerformed(ActionEvent ae)
   {
      if (ae.getSource() == bEx1) {
         AcceptNameAWT dialog1 = new AcceptNameAWT("Form clicked by
open button 1");
         dialog1.pack();
         dialog1.show();
      }
      else if (ae.getSource() == bEx2) {
         TextComments dialog2 = new TextComments("Form clicked by open
button 2");
         dialog2.setSize(200, 200);
         dialog2.show();
      }
      else {
         JOptionPane.showMessageDialog(this, "See you again");
         System.exit(0);
      }
   }
    public static void main(String args[])
   {
      ActionHandler ObjStar = new ActionHandler("Action handling in
Java");
      ObjStar.setSize(400,200);
      ObjStar.show();
   }
}
```

The result:



6. This sample demonstrates how to use Radio button group and capture selection event. on them.



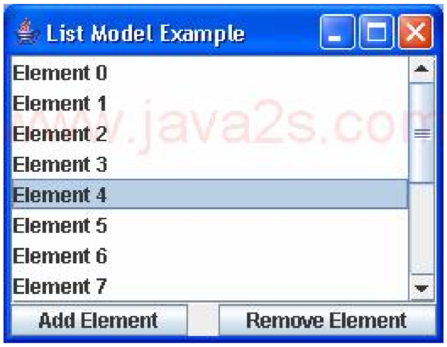
```
package components;

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

/*
 * RadioButtonDemo.java requires these files:
 *   images/Bird.gif
 *   images/Cat.gif
 *   images/Dog.gif
 *   images/Rabbit.gif
 *   images/Pig.gif
 */
public class RadioButtonDemo extends JPanel
                             implements ActionListener {
    static String birdString = "Bird";
    static String catString = "Cat";
    static String dogString = "Dog";
    static String rabbitString = "Rabbit";
    static String pigString = "Pig";

    JLabel picture;

    public RadioButtonDemo() {
        super(new BorderLayout());

        //Create the radio buttons.
        JRadioButton birdButton = new JRadioButton(birdString);
        birdButton.setMnemonic(KeyEvent.VK_B);
        birdButton.setActionCommand(birdString);
        birdButton.setSelected(true);

        JRadioButton catButton = new JRadioButton(catString);
        catButton.setMnemonic(KeyEvent.VK_C);
        catButton.setActionCommand(catString);
        JRadioButton dogButton = new JRadioButton(dogString);
        dogButton.setMnemonic(KeyEvent.VK_D);
        dogButton.setActionCommand(dogString);

        JRadioButton rabbitButton = new JRadioButton(rabbitString);
        rabbitButton.setMnemonic(KeyEvent.VK_R);
        rabbitButton.setActionCommand(rabbitString);

        JRadioButton pigButton = new JRadioButton(pigString);
        pigButton.setMnemonic(KeyEvent.VK_P);
        pigButton.setActionCommand(pigString);

        //Group the radio buttons.
        ButtonGroup group = new ButtonGroup();
        group.add(birdButton);
        group.add(catButton);
        group.add(dogButton);
        group.add(rabbitButton);
        group.add(pigButton);

        //Register a listener for the radio buttons.
        birdButton.addActionListener(this);
        catButton.addActionListener(this);
        dogButton.addActionListener(this);
        rabbitButton.addActionListener(this);
        pigButton.addActionListener(this);

        //Set up the picture label.
        picture = new JLabel(createImageIcon("images/"
                                             + birdString
                                             + ".gif"));

        //The preferred size is hard-coded to be the width of the
        //widest image and the height of the tallest image.
        //A real program would compute this.
        picture.setPreferredSize(new Dimension(177, 122));


        //Put the radio buttons in a column in a panel.
        JPanel radioPanel = new JPanel(new GridLayout(0, 1));
        radioPanel.add(birdButton);
        radioPanel.add(catButton);
        radioPanel.add(dogButton);
        radioPanel.add(rabbitButton);
        radioPanel.add(pigButton);

        add(radioPanel, BorderLayout.LINE_START);
        add(picture, BorderLayout.CENTER);
        setBorder(BorderFactory.createEmptyBorder(20,20,20,20));
    }

    /** Listens to the radio buttons. */
    public void actionPerformed(ActionEvent e) {
        picture.setIcon(createImageIcon("images/"
                                        + e.getActionCommand()
                                        + ".gif"));
    }

    /** Returns an ImageIcon, or null if the path was invalid. */
    protected static ImageIcon createImageIcon(String path) {
        java.net.URL imgURL = RadioButtonDemo.class.getResource(path);
        if (imgURL != null) {
            return new ImageIcon(imgURL);
        } else {
            System.err.println("Couldn't find file: " + path);
            return null;
        }
    }

    /**
     * Create the GUI and show it.  For thread safety,
     * this method should be invoked from the
     * event-dispatching thread.
     */
    private static void createAndShowGUI() {
        //Create and set up the window.
        JFrame frame = new JFrame("RadioButtonDemo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Create and set up the content pane.
        JComponent newContentPane = new RadioButtonDemo();
        newContentPane.setOpaque(true); //content panes must be opaque
        frame.setContentPane(newContentPane);

        //Display the window.
        frame.pack();
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        //Schedule a job for the event-dispatching thread:
        //creating and showing this application's GUI.
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }
}
```

7. This sample tutors you how to create a list and response to item selection action.

```
/*
Java Swing, 2nd Edition
By Marc Loy, Robert Eckstein, Dave Wood, James Elliott, Brian Cole
ISBN: 0-596-00408-7
Publisher: O'Reilly
```

```
*/

// ListModelExample.java
//An example of JList with a DefaultListModel that we build up at runtime.
//

import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.DefaultListModel;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JPanel;
import javax.swing.JScrollPane;

public class ListModelExample extends JPanel {

  JList list;

  DefaultListModel model;

  int counter = 15;

  public ListModelExample() {
    setLayout(new BorderLayout());
    model = new DefaultListModel();
    list = new JList(model);
    JScrollPane pane = new JScrollPane(list);
    JButton addButton = new JButton("Add Element");
    JButton removeButton = new JButton("Remove Element");
    for (int i = 0; i < 15; i++)
      model.addElement("Element " + i);

    addButton.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        model.addElement("Element " + counter);
        counter++;
      }
    });
    removeButton.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        if (model.getSize() > 0)
          model.removeElementAt(0);
      }
    });

    add(pane, BorderLayout.NORTH);
    add(addButton, BorderLayout.WEST);
    add(removeButton, BorderLayout.EAST);
  }

  public static void main(String s[]) {
    JFrame frame = new JFrame("List Model Example");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setContentPane(new ListModelExample());
    frame.setSize(260, 200);
    frame.setVisible(true);
  }
}
```

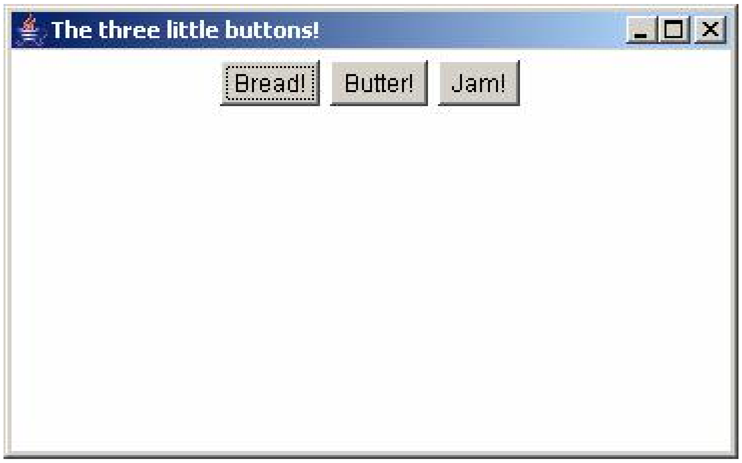
The result:



Do It Yourself

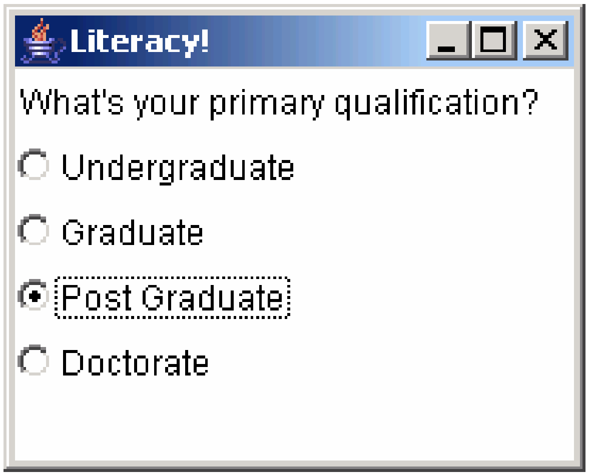
1.1. Do workshop 1, 2

1.2. Write a program that have the main window with three buttons



When clicking on any button, title of that button should be displayed as a message dialog.

1.3. Write a program that have the main window with radio buttons. You should refer to the above exercise 3



What's your primary qualification?

○ Undergraduate

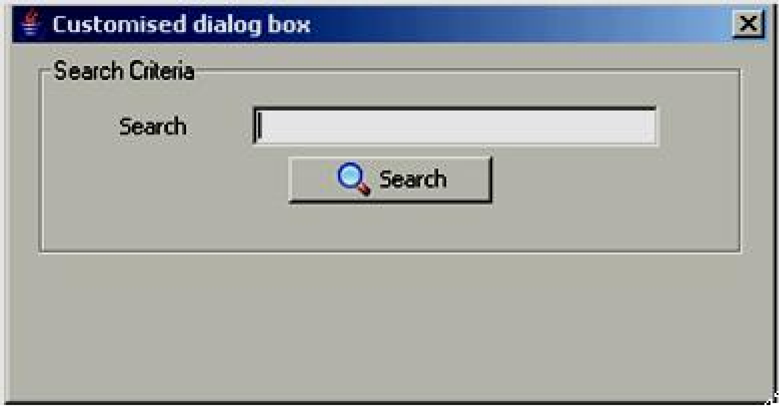
○ Graduate

● Post Graduate

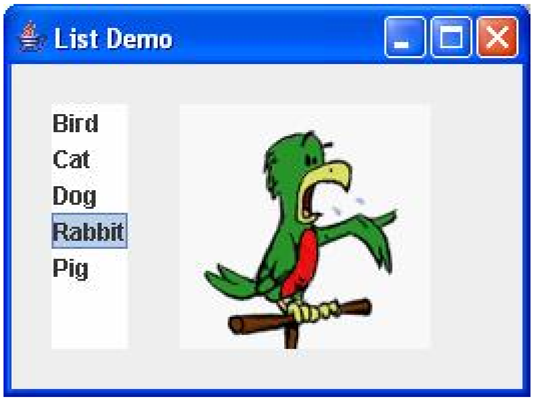
○ Doctorate

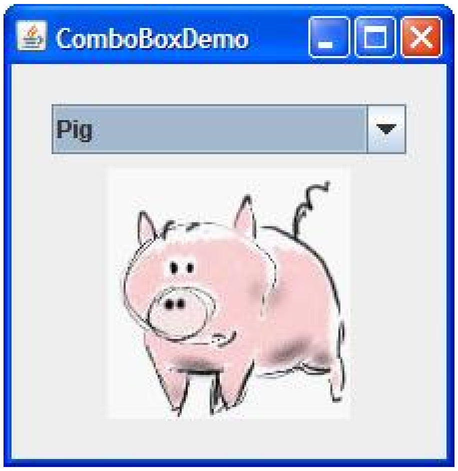
Display a message dialog showing the selected radio item.

1.4. Create your own dialog box by extending JDialog



1.5. Try to program a application like the sample 6, but let’s use a list or a combo box.



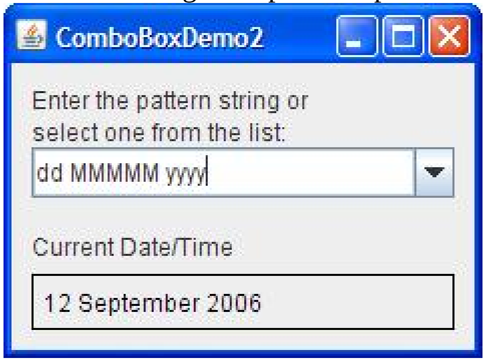


1.6. Making the program in the exercise 1.4 running on a web page.

Self-study Samples

+ Combo box

/tutorial/uiswing/examples/components/index.html#ComboBoxDemo2



Enter the pattern string or select one from the list:

dd MMMMM yyyy

Current Date/Time

12 September 2006

+ File Chooser

/tutorial/uiswing/examples/components/index.html#FileChooserDemo

+ Java Tutorial

+ java2s.com

+ javapassion.com

Short Useful Code Snippets

1. Create a checkbox with mnemonic key

```
    chinButton = new JCheckBox("Chin");
    chinButton.setMnemonic(KeyEvent.VK_C);
    chinButton.setSelected(true);
```

2. Capture the selected checkbox button

```
public void itemStateChanged(ItemEvent e) {
    ...
    Object source = e.getItemSelectable();

    if (source == chinButton) {
        //...make a note of it...
    } else if (source == glassesButton) {
        //...make a note of it...
    } else if (source == hairButton) {
        //...make a note of it...
    } else if (source == teethButton) {
        //...make a note of it...
    }

    if (e.getStateChange() == ItemEvent.DESELECTED)
        //...make a note of it...
    ...
}
```

3. Make some radio buttons into a group

```
    //Group the radio buttons.
    ButtonGroup group = new ButtonGroup();
    group.add(birdButton);
    group.add(catButton);
    group.add(dogButton);
```

4. Capture event when another list item selected

```
public class ComboBoxDemo ... implements ActionListener {
    . . .
        petList.addActionListener(this) {
    . . .
    public void actionPerformed(ActionEvent e) {
        JComboBox cb = (JComboBox)e.getSource();
        String petName = (String)cb.getSelectedItem();
        updateLabel(petName);
    }
    . . .
}
```

5. Create a combo-box from a predefined array.

```
String[] patternExamples = {
         "dd MMMMM yyyy",
         "dd.MM.yy",
         "MM/dd/yy",
         "yyyy.MM.dd G 'at' hh:mm:ss z",
         "EEE, MMM d, ''yy",
         "h:mm a",
         "H:mm:ss:SSS",
         "K:mm a,z",
         "yyyy.MMMMM.dd GGG hh:mm aaa"
};
. . .
JComboBox patternList = new JComboBox(patternExamples);
```

6. Wrap a component by a scrollpane.

```
//Put the editor pane in a scroll pane.
```

```
JScrollPane editorScrollPane = new JScrollPane(editorPane);
editorScrollPane.setVerticalScrollBarPolicy(
                JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
editorScrollPane.setPreferredSize(new Dimension(250, 145));
editorScrollPane.setMinimumSize(new Dimension(10, 10));
….
// Add scrollpane into the container
cp.add(editorScrollPane );
```

7. Load a webpage in JEditorPane

```
JEditorPane editorPane = new JEditorPane();
        editorPane.setEditable(false);
        java.net.URL helpURL = CurrentClassName.class.getResource(
                                        "somepage.html");
        if (helpURL != null) {
            try {
                editorPane.setPage(helpURL);
            } catch (IOException e) {
                System.err.println("Attempted to read a bad URL: " +
helpURL);
            }
        } else {
            System.err.println("Couldn't find file:
TextSampleDemoHelp.html");
        }
```