Chapter-2.1
User interface components with AWT

User Interface

- A user interface is a type of software that enables an application to interact with a user. There are two types of user interface.
    1. Character user interface (CUI)
    2. Graphical user interface(GUI)

CUI vs GUI

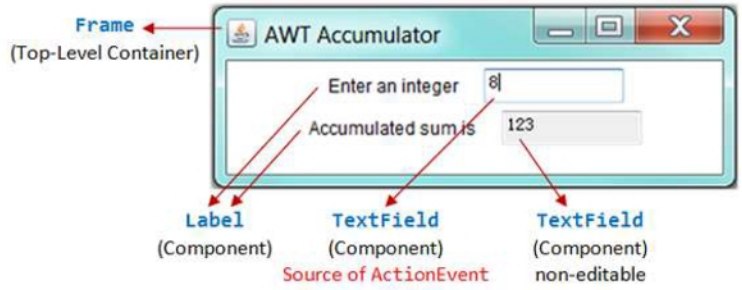
CUI	GUI
1. CUI is the Character User Interface.	1. GUI is Graphical User Interface.
2. DOS is the characterized as CUI.	2. Windows is characterized as GUI.
3. A CUI use characters on screen that control with keyboard.	3. A GUI use pictures, symbols, word that control with mouse.
4. CUI can be confusing & difficult to remember.	4. GUI is very friendly & easy to remember.

Advantage of GUI over CUI

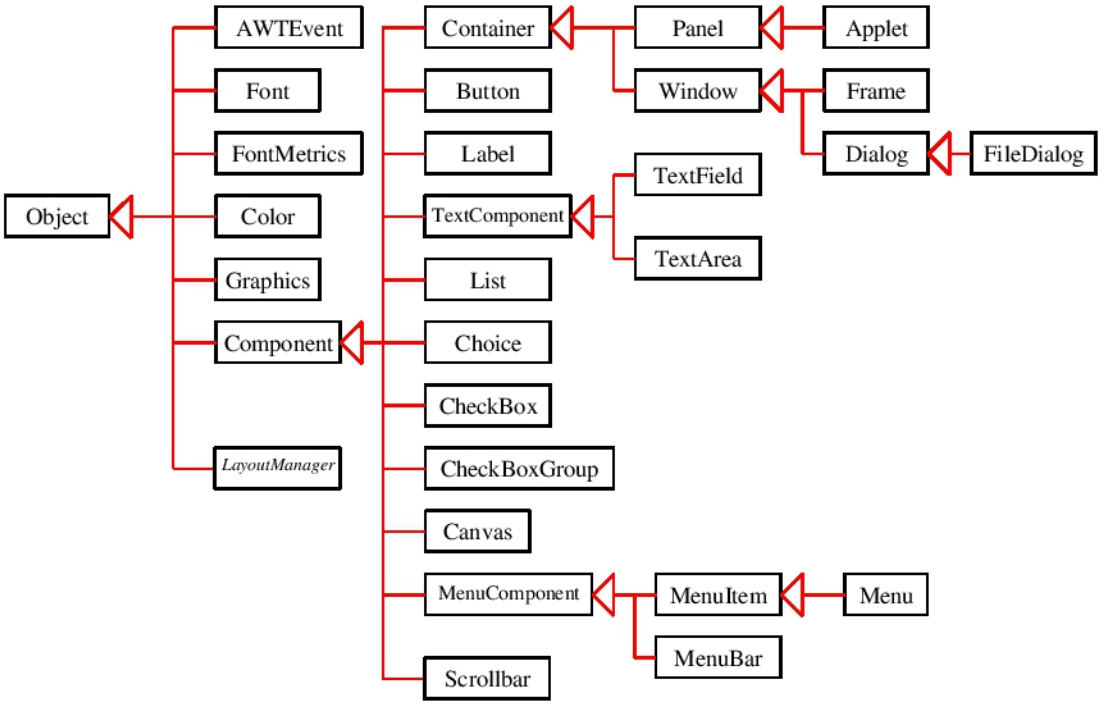
1. GUI provides graphical icons to interact while the CUI (Character User Interface) offers the simple text-based interfaces.
2. GUI makes the application more entertaining and interesting on the other hand CUI does not.
3. GUI offers click and execute environment while in CUI every time we have to enter the command for a task.
4. New user can easily interact with graphical user interface by the visual indicators but it is difficult in Character user interface.
5. Windows concept in GUI allow the user to view, manipulate and control the multiple applications at once while in CUI user can control one task at a time.
6. GUI provides multitasking environment so as the CUI also does but CUI does not provide same ease as the GUI do.
7. Using GUI, it is easier to control and navigate the operating system which becomes very slow in command user interface. GUI can be easily customized.

Introduction to Abstract Window Toolkit (AWT)

- Abstract Window Toolkit (AWT) is a set of application program interfaces (API s) used by Java programmers to create graphical user interface (GUI) objects, such as buttons, scroll bars, and windows.
- The most widely used AWT package are **java.awt** and **java.awt.event.**
- The java.awt package contains the core AWT graphics classes as listed below.
    - i. GUI component classes such as Button, TextField, Label etc.
    - ii. GUI container classes such as Frame, Panel, Dialog etc.
    - iii. Layout Manager such as FlowLayout, GridLayout etc.
    - iv. Custom graphics classes such as Color, Font etc.
- The java.awt.event package provides event handling
    - i. Event classes such as MouseEvent, KeyEvent, WindowEvent etc.
    - ii. Event Listener interfaces such as MouseListener, KeyListener, WindowListener etc.
    - iii. Event Listener Adapter class such as MouseAdapter, KeyAdapter, WindowAdapter etc.
- The below figure shows different frame as container consisting of several GUI components.



The following diagram shows the diagram of the classes of AWT package.



GUI Class Hierarchy (AWT)

Features of java AWT

1. A rich set of user interface component.
2. A robust event handling model.
3. Graphics and imaging tools such as different shape, color.
4. Layout manager, for flexible window layouts that do not depend on a particular window size or screen resolution.
5. Data transfer classes, for cut and paste through the native platform clipboard.

Component

- Component is an object having a graphical representation that can be displayed on the screen and that can interact with the user.
- For examples buttons, checkboxes, list and scrollbars of a graphical user interface.

Container

- Container object is components that can contain other components.
- The container is subclass of Component.
- Every GUI program has a top level container in AWT such as Window, Frame, Dialog, Applet etc.

Window

- The window class creates top level container.
- Generally, we don't create window object directly but instead we use a subclass of window called **Frame**.

Frame

- A Frame is a top-level window with a title and a border.
- The size of the frame includes any area designated for the border.
- Frame encapsulates window. It has a title bar, menu bar, borders, and resizing corners.
- The two constructor for Frame are
  1. public Frame()
  2. public Frame(String title)

Some of the methods in Frame class are

1. public String getTitle ()
   The getTitle() method returns the current title for the Frame. If there is no title, this method returns null.
2. public void setTitle (String title)
   The setTitle() method changes the Frame's title to title.
3. public Image getIconImage ()
   The getIconImage() method returns the image used as the icon. Initially, this returns null. For some platforms, the method should not be used because the platform does not support the concept.
4. public void setIconImage (Image image)
5. The setIconImage() method changes the image to display when the Frame is iconified to image. Not all platforms utilize this resource.
6. public boolean isResizable ()
   The isResizable() method will tell you if the current Frame is resizable.
7. public void setResizable (boolean resizable)
   The setResizable() method changes the resize state of the Frame. A resizable value of true means the user can resize the Frame, false means the user cannot. This must be set before the Frame is shown or the peer created.
8. public void setCursor (int cursorType)
   The setCursor() method changes the cursor of the Frame to cursorType. cursorType must be one of the cursor constants provided with the Frame class. If cursorType is not one of the predefined cursor types, setCursor() throws the IllegalArgumentException run-time exception.
9. public void setSize(int width, int height)
   Set the frame size to specified size.
10. void show(): display the current frame
11. void hide(): hides the current frame
12. void setUndecorated(boolean b):Removes title bar and boundary of frame if b is true.
13. void setVisible(boolean b): show or hide the frame based on specified Boolean value.
    Note: show() and hide () are Deprecated. As of JDK version 1.5, replaced by setVisible(boolean b).

    We can set following types of cursor

    public final static int DEFAULT_CURSOR
    public final static int CROSSHAIR_CURSOR
    public final static int TEXT_CURSOR
    public final static int WAIT_CURSOR
    public final static int SW_RESIZE_CURSOR
    public final static int SE_RESIZE_CURSOR

public final static int NW_RESIZE_CURSOR
public final static int NE_RESIZE_CURSOR
public final static int N_RESIZE_CURSOR
public final static int S_RESIZE_CURSOR
public final static int W_RESIZE_CURSOR
public final static int E_RESIZE_CURSOR
public final static int HAND_CURSOR
public final static int MOVE_CURSOR

14. public int getCursorType ()
    The getCursorType() method retrieves the current cursor.

Example:
Program to display a frame in different ways.

Way1:

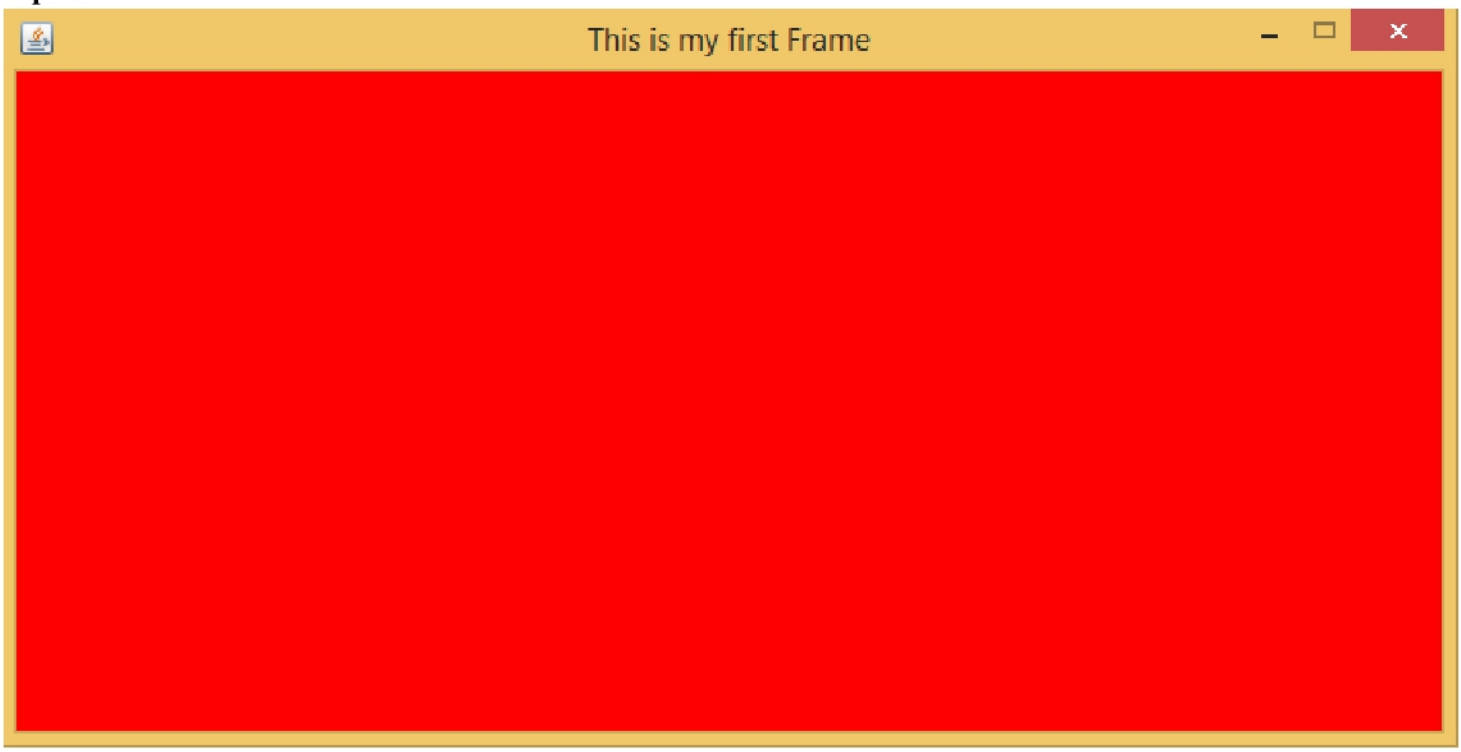
```
//import java.awt.Color;
//import java.awt.Cursor;
//import java.awt.Frame;

//or simply use

import java.awt.*;

public class Demo
{
	public static void main(String [] args)
	{
		Frame fm=new Frame("This is my first Frame"); //set title bar text
		fm.setSize(700,350); //sett size of frame
		fm.setLocationRelativeTo(null); // set frame position as center of window(optional)
		fm.setResizable(false); // set the frame as non resizable(optional)
		fm.setBackground(Color.RED); //set background color to red (optional)
		fm.setLayout(null); // use no layout manager (optional)
		fm.setCursor(new Cursor(Cursor.HAND_CURSOR)); // set cursor(optional)
		fm.show();
	}
}
```

Way2:

```
import java.awt.*;
public class Demo extends Frame
{
	public Demo()
	{
		super("This is my first Frame"); //must be the first statement
		setSize(700,350);
		setLocationRelativeTo(null);
		setResizable(false);
		setBackground(Color.RED);
		setLayout(null);
		setCursor(new Cursor(Cursor.HAND_CURSOR));
		show();
	}
```

```
        public static void main(String [] args) throws Exception
        {
                new Demo();
        }
}
```

Output:



Note:

super keyword can also be used to access the parent class constructor.
Example:

```
class Parent
{
        public Parent()
        {
                System.out.println("Hello World");
        }
}

class Child extends Parent
{
        public Child()
        {
                super();
        }

}
public class Demo
{
        public static void main(String [] args)
        {
                Child c=new Child();
        }
}
```

```
C:\Users\Bheeshma\Desktop>java Demo
Hello World
```

Panel

- Panel is a rectangular box placed inside a top level container which provides a place for holding collections of other components such as buttons, labels, textfields etc.
- A panel is a window that doesn't contain title bar, menu bar or boarder.
- It is simply a subclass of Container class.

The constructor for Panel is

public panel()

Some of the methods are

1. add(Component c) : it is used to add other components inside panel.
2. setSize(int width, int height) : it is used to define the custom size of panel.
3. setLocation(int x, int y): it is used to define the location of panel.
4. setBounds(int x, int y, int width, int height) : it is used to set the position and boundary size of the panel.
5. setBackground(Color) : it is used to set the background color of panel.

Example:

```
import java.awt.*;
class Demo extends Frame
{
        Panel pTop;

        Panel upper;
        Panel pGender;
        Panel pMarital;



        Panel pMiddle;
        Panel pBottom;

        public Demo()
          {
              super("This is my first Frame");
              setSize(700,350);
              setBackground(Color.RED);
              setLayout(null);

              pTop=new Panel();
              pTop.setLayout(null);
              pTop.setBounds(10,50,680,290);
              pTop.setBackground(Color.GREEN);

              upper =new Panel();
              upper.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
              upper.setBackground(Color.WHITE);
              upper.setBounds(5,5,670,30);
              pTop.add(upper);

              pGender =new Panel();
              pGender.setBounds(5,35,670,30);
```

```
                pGender.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pGender.setBackground(Color.WHITE);
                pTop.add(pGender);


                pMarital =new Panel();
                pMarital.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pMarital.setBounds(5,65,670,30);
                pMarital.setBackground(Color.WHITE);
                pTop.add(pMarital);

                pMiddle=new Panel();
                pMiddle.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pMiddle.setBounds(5,90,670,30);
                pMiddle.setBackground(Color.RED);
                pTop.add(pMiddle);

                pBottom=new Panel();
                pBottom.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pBottom.setBounds(5,130,670,100);
                pBottom.setBackground(Color.WHITE);
                pTop.add(pBottom);

                add(pTop);

                show();
            }
        public static void main(String [] str)
        {
                new Demo();
        }
}
```

Output:

Menu Bars and Menu

- The top level window can have **menu bar** associated with it.
- **Menus** can exist only in the menu bars, and menu bars can be attached only to Frames.
- Each item in menu is represented by a **MenuItem** object.
- When the user clicks a menu item, it generates an **Action** event.
- The constructor for creating Menu Bar is

  public MenuBar()

- Some of the methods are
  - Menu add(Menu m) : Adds the specified menu to the menu bar.
  - void remove(int index) :Removes the menu located at the specified index from this menu bar.

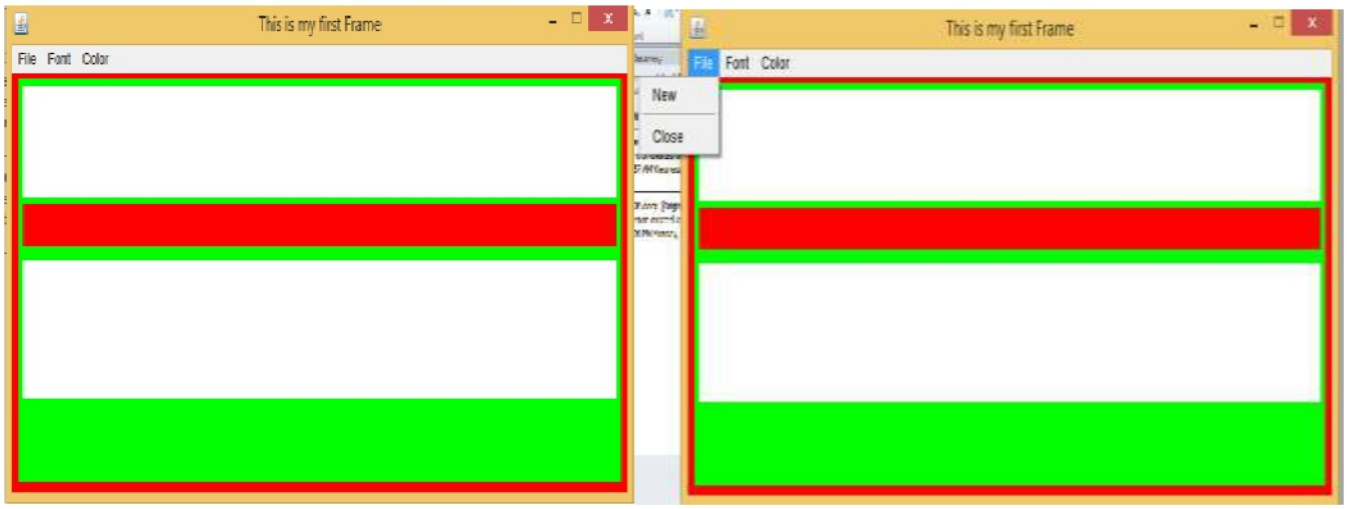
Example:

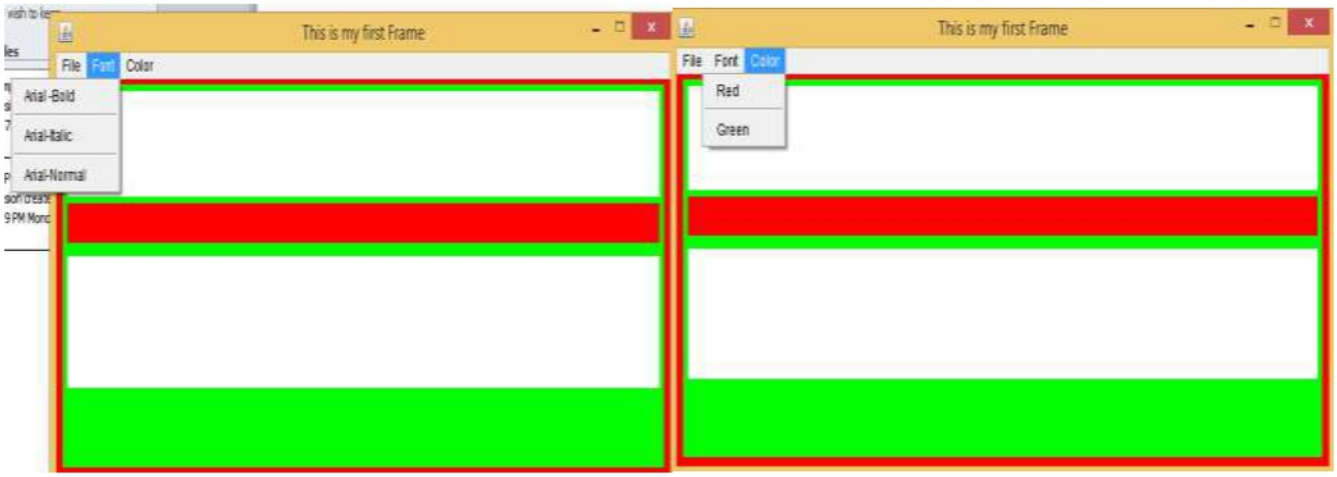
```
import java.awt.*;
class Demo extends Frame
{
        MenuBar mb;
        Menu mFile,mFont,mColor;
        MenuItem miNew,miClose,miSeparator1;
        MenuItem miFont1,miSeparator2,miFont2,miFont3,miSeparator3;
        MenuItem miRed,miSeparator4,miGreen;
        Panel pTop;
        Panel pMiddle;
        Panel upper;
        Panel pBottom;
        Panel pGender;
        Panel pMarital;

        public Demo()
          {
              super("This is my first Frame");
              setSize(700,350);
              setBackground(Color.RED);
```

```
setLayout(null);

mb=new MenuBar();

mFile=new Menu("File");
mFont=new Menu("Font");
mColor=new Menu("Color");

mb.add(mFile);
mb.add(mFont);
mb.add(mColor);

miNew=new  MenuItem("New");
miSeparator1=new  MenuItem("-");
miClose=new MenuItem("Close");

mFile.add(miNew);
mFile.add(miSeparator1);
mFile.add(miClose);

miFont1=new  MenuItem("Arial -Bold");
miSeparator2=new  MenuItem("-");
miFont2=new MenuItem("Arial-Italic");
miFont3=new MenuItem("Arial-Normal");
miSeparator3=new  MenuItem("-");

mFont.add(miFont1);
mFont.add(miSeparator2);
mFont.add(miFont2);
mFont.add(miSeparator3);
mFont.add(miFont3);

miRed=new  MenuItem("Red");
miSeparator4=new  MenuItem("-");
miGreen=new MenuItem("Green");

mColor.add(miRed);
mColor.add(miSeparator4);
mColor.add(miGreen);


setMenuBar(mb);  //ADD MENUBAR TO FRAME



pTop=new Panel();
pTop.setLayout(null);
pTop.setBounds(10,50,680,290);
pTop.setBackground(Color.GREEN);


upper =new Panel();
upper.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
```

```
                upper.setBackground(Color.WHITE);
                upper.setBounds(5,5,670,30);
                pTop.add(upper);

                pGender =new Panel();
                pGender.setBounds(5,35,670,30);
                pGender.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pGender.setBackground(Color.WHITE);
                pTop.add(pGender);

                pMarital =new Panel();
                pMarital.setBounds(5,65,670,30);
                pMarital.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pMarital.setBackground(Color.WHITE);
                pTop.add(pMarital);

                pMiddle=new Panel();
                pMiddle.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pMiddle.setBounds(5,90,670,30);
                pMiddle.setBackground(Color.RED);
                pTop.add(pMiddle);

                pBottom=new Panel();
                pBottom.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pBottom.setBounds(5,130,670,100);
                pBottom.setBackground(Color.WHITE);
                pTop.add(pBottom);

                add(pTop);
                show();
            }

        public static void main(String [] str)
        {
                new Demo();
        }

    }
```

Output:





Choice

- Choice control is used to show pop up menu of choices.
- Selected choice is shown on the top of the menu.
- When the user chooses an item, the Choice generates an **Action** event.
- The constructor that is used to create choice is

Choice ch=new Choice();

Some of the methods that are used in choice are

1. addItem(String item): add the items in a Choice List
2. String getSelectedItem(): Returns the selected item in string
3. int countItems(): Returns the number of items in Choice (**Deprecated.**)
4. int getItemCount() :Returns the number of items in this Choice menu.
5. int getSelectedIndex() :Returns the index of the currently selected item.
6. void insert(String item, int index) :Inserts the item into this choice at the specified position.
7. void remove(int position) :Removes an item from the choice menu at the specified position.
8. void remove(String item) :Removes the first occurrence of item from the Choice menu.
9. void removeAll() :Removes all items from the choice menu.
10. void select(int pos):Sets the selected item in this Choice menu to be the item at the specified position.
11. void select(String str) :Sets the selected item in this Choice menu to be the item whose name is equal to the specified string

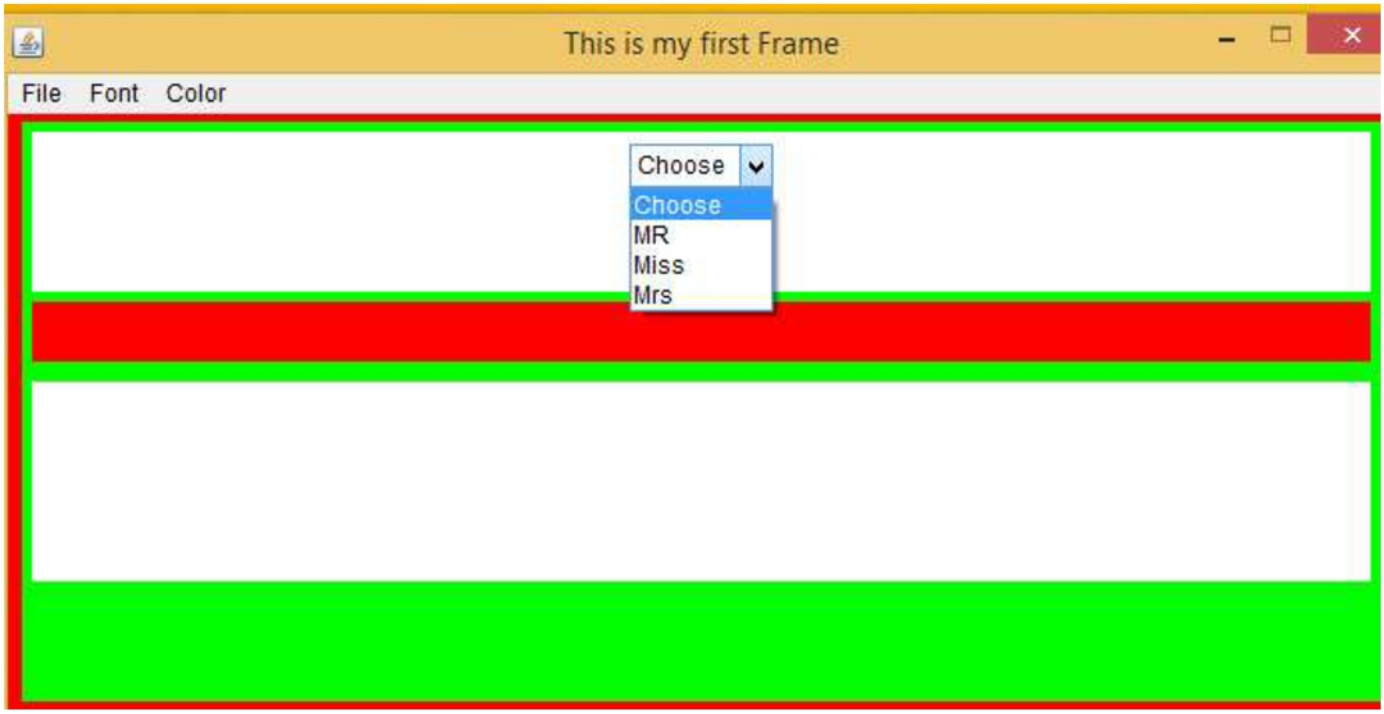
Example:

```
import java.awt.*;
class Demo extends Frame
{
        MenuBar mb;
        Menu mFile,mFont,mColor;
        MenuItem miNew,miClose,miSeparator1;
        MenuItem miFont1,miSeparator2,miFont2,miFont3,miSeparator3;
        MenuItem miRed,miSeparator4,miGreen;
        Panel pTop;
        Panel pMiddle;
        Panel upper;
        Panel pBottom;
        Panel pGender;
        Panel pMarital;
        Choice ch;
        public Demo()
        {
                super("This is my first Frame");
                setSize(700,350);//700
```

```
setBackground(Color.RED);
setLayout(null);

mb=new MenuBar();
setMenuBar(mb);
mFile=new Menu("File");
mFont=new Menu("Font");
mColor=new Menu("Color");
mb.add(mFile);
mb.add(mFont);
mb.add(mColor);
miNew=new  MenuItem("New");
miSeparator1=new  MenuItem("-");
miClose=new MenuItem("Close");
mFile.add(miNew);
mFile.add(miSeparator1);
mFile.add(miClose);
miFont1=new  MenuItem("Arial -Bold");
miSeparator2=new  MenuItem("-");
miFont2=new MenuItem("Arial-Italic");
miFont3=new MenuItem("Arial-Normal");
miSeparator3=new  MenuItem("-");
mFont.add(miFont1);
mFont.add(miSeparator2);
mFont.add(miFont2);
mFont.add(miSeparator3);
mFont.add(miFont3);
miRed=new  MenuItem("Red");
miSeparator4=new  MenuItem("-");// seperator
miGreen=new MenuItem("Green");
mColor.add(miRed);
mColor.add(miSeparator4);
mColor.add(miGreen);



pTop=new Panel();
pTop.setLayout(null);
pTop.setBounds(10,50,680,290);//640
pTop.setBackground(Color.GREEN);


upper =new Panel();
upper.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
upper.setBackground(Color.WHITE);
upper.setBounds(5,5,670,30);


ch=new Choice();
ch.addItem("Choose");
ch.addItem("MR");
ch.addItem("Miss");
ch.addItem("Mrs");
```

```
                upper.add(ch);


                pTop.add(upper);

                pGender =new Panel();
                pGender.setBounds(5,35,670,30);
                pGender.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pGender.setBackground(Color.WHITE);
                pTop.add(pGender);


                pMarital =new Panel();
                pMarital.setBounds(5,65,670,30);
                pMarital.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pMarital.setBackground(Color.WHITE);
                pTop.add(pMarital);

                pMiddle=new Panel();
                pMiddle.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pMiddle.setBounds(5,90,670,30);
                pMiddle.setBackground(Color.RED);
                pTop.add(pMiddle);

                pBottom=new Panel();
                pBottom.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pBottom.setBounds(5,130,670,100);
                pBottom.setBackground(Color.WHITE);
                pTop.add(pBottom);

                add(pTop);
                show();
        }
        public static void main(String [] str)
        {
                new Demo();
        }
}
```

Output:



Label

- The label class is used to put un selectable text in our GUI form. The constructors used to create label are
  Label lbl=new Label();
  Label lbl1=new Label(String txt);
  Label lbl2=new Label(String txt, int Alignment); where Alignments can be LEFT, RIGHT or CENTER
  Eg. Label a=new Label("Name",Label.RIGHT)
- The Label class has the following methods
  1. String getText(): Returns the Label's text.
  2. void setText(String txt); Set the Label's text.
  3. Int getAlignment(): Returns the alignment of the Label
  4. setAlignment(int allignment); Set the alignment of the Label.

TextField

- A Textfield used for taking single line string input from user. The following constructors are used to create TextField

Name	Description
`TextField()`	This constructor creates instance of a Textfield.
`TextField(int columns)`	This constructor creates instance of a Textfield with the given number of Columns
`TextField(String text)`	This constructor creates instance of a Textfield with the given text.
`TextField(String text, int columns)`	This constructor creates instance of a Textfield with the given text and number of columns.

The TextField class has the following methods.

1. void setText(String text): Sets the TextField text.
2. String getText(): Returns the text from the TextField
3. void setEchoChar(Char ch): Sets the Echo Character for password privacy
4. char getEchoChar(): Returns the echo character from the TextField
5. boolean echoCharIsSet() : Returns true if echo charter is set in the TextField

6. int getColumns():Returns the number of column from the TextField
7. void setColumns(int number): Sets the given number of columns in the TextField
8. void setEditable(boolean state): sets the specified boolean to indicate whether or not this textfield should be editable.
9. boolena isEditable() : Return true if this textfield is editable otherwise false

TextArea

- A TextArea used for taking input from user.
- TextArea have **multiple line region** to get input from user i.e. TextArea class is used to take multiline input text.
- The following constructors are used to create TextArea.

Name	Description
`TextArea()`	This constructor creates instance of a TextArea.
`TextArea(int rows, int columns)`	This constructor creates instance of a Textarea with the given number of Columns and row.
`TextArea(String text)`	Constructs a new text area with the specified text.
`TextArea(String text, int rows, int columns)`	Constructs a new text area with the specified text and also row and column.
`TextArea(String text, int rows, int columns, int scrollbars)`	Constructs a new text area with the specified text,row and column and scrollbar.

The scrollbars can be any of the following
SCROLLBAR_BOTH, SCROLLBARS_NONE, SCROLLBARS_HORIZONTAL_ONLY, SCROLLBARS_VERTICAL-ONLY

Example: TextArea ab=new TextArea(" ",30,25,TextArea.SCROLLBARS_BOTH);

The TextArea class has the following methods.

1. setText(String str): Sets the TextArea text
2. getText(): Returns the Text in TextArea
3. appendText(String str): Append String str at the end

Button

- Button class use to perform action when the Button is clicked.
- The button is a component that contain a label and generates an event when it is pressed.
- The following constructors are used to create Button

Name	Description
`Button()`	This constructor creates instance of a Button.
`Button(String Text)`	This constructor creates instance of a Button with the specified text.

- The Button class has the following methods.
  1. setLabel(String str): Set the label to a button
  2. getLabel(): Returns the label as string

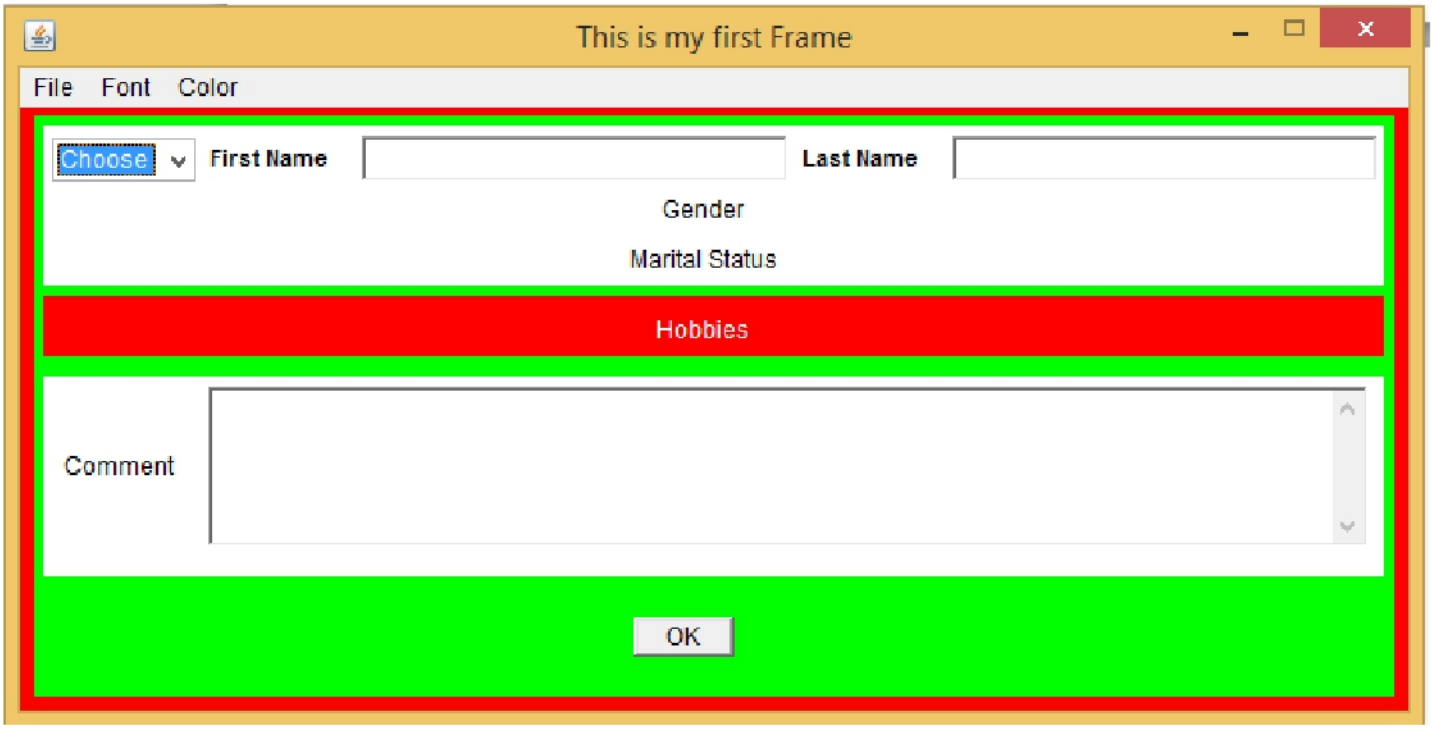
Example:

```
import java.awt.*;
class Demo extends Frame
{
        Button btn_OK;
        MenuBar mb;
        Menu mFile,mFont,mColor;
        MenuItem miNew,miClose,miSeparator1;
        MenuItem miFont1,miSeparator2,miFont2,miFont3,miSeparator3;
        MenuItem miRed,miSeparator4,miGreen;
        Panel pTop;
        Panel pMiddle;
        Panel upper;
        Panel pBottom;
        Panel pGender;
        Panel pMarital;

        Label lbl_maritalstatus;
        Label lblcmt;
        Label lblhoby;
        Label lbl_name;
        Label lbl_lastname;
        Label lbl_Gender;
        Choice ch;
        TextField txtname;
        TextField txtln;
        TextArea txtcmt;

        Font fon=new Font("Arial",Font.BOLD,11);
        Font fon1=new Font("Arial",Font.ITALIC,11);
        Font fon0=new Font("Arial",Font.PLAIN,11);

        public Demo()
        {
                super("This is my first Frame");
                setSize(700,350);
                setLocationRelativeTo(null);
                setResizable(false);
                setBackground(Color.RED);
                setLayout(null);
                setCursor(new Cursor(Cursor.HAND_CURSOR));

                mb=new MenuBar();
                setMenuBar(mb);
                mFile=new Menu("File");
                mFont=new Menu("Font");
                mColor=new Menu("Color");
                mb.add(mFile);
                mb.add(mFont);
                mb.add(mColor);
                miNew=new  MenuItem("New");
                miSeparator1=new  MenuItem("-");
                miClose=new MenuItem("Close");
```

```
mFile.add(miNew);
mFile.add(miSeparator1);
mFile.add(miClose);
miFont1=new  MenuItem("Arial -Bold");
miSeparator2=new  MenuItem("-");
miFont2=new MenuItem("Arial-Italic");
miFont3=new MenuItem("Arial-Normal");
miSeparator3=new  MenuItem("-");
mFont.add(miFont1);
mFont.add(miSeparator2);
mFont.add(miFont2);
mFont.add(miSeparator3);
mFont.add(miFont3);
miRed=new  MenuItem("Red");
miSeparator4=new  MenuItem("-");
miGreen=new MenuItem("Green");
mColor.add(miRed);
mColor.add(miSeparator4);
mColor.add(miGreen);

pTop=new Panel();
pTop.setLayout(null);
pTop.setBounds(10,50,680,290);
pTop.setBackground(Color.GREEN);

upper =new Panel();
upper.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
upper.setBackground(Color.WHITE);
upper.setBounds(5,5,670,30);
ch=new Choice();
ch.addItem("Choose");
ch.addItem("MR");
ch.addItem("Miss");
ch.addItem("Mrs");

lbl_name=new Label("First Name");
lbl_name.setFont(fon);
txtname=new TextField(27);
lbl_lastname=new Label("Last Name");
lbl_lastname.setFont(fon);
txtln=new TextField(27);

upper.add(ch);
upper.add(lbl_name);
upper.add(txtname);
upper.add(lbl_lastname);
upper.add(txtln);

pTop.add(upper);
pGender =new Panel();
pGender.setBounds(5,35,670,30);
pGender.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
pGender.setBackground(Color.WHITE);
```

```
                lbl_Gender=new Label("Gender");
                pGender.add(lbl_Gender);
                pTop.add(pGender);


                pMarital =new Panel();
                pMarital.setBounds(5,65,670,30);
                pMarital.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pMarital.setBackground(Color.WHITE);
                lbl_maritalstatus=new Label("Marital Status");
                pMarital.add(lbl_maritalstatus);
                pTop.add(pMarital);

                pMiddle=new Panel();
                pMiddle.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pMiddle.setBounds(5,90,670,30);
                pMiddle.setBackground(Color.RED);
                lblhoby=new Label("Hobbies");
                lblhoby.setForeground(Color.WHITE);
                pMiddle.add(lblhoby);
                pTop.add(pMiddle);

                pBottom=new Panel();
                pBottom.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pBottom.setBounds(5,130,670,100);
                pBottom.setBackground(Color.WHITE);
                lblcmt=new Label("Comment");
                pBottom.add(lblcmt);
                txtcmt=new TextArea(4,80);
                pBottom.add(txtcmt);
                pTop.add(pBottom);

                btn_OK=new Button("OK");
                btn_OK.setBounds(300,250,50,20);
                pTop.add(btn_OK);

                add(pTop);
                show();
        }


        public static void main(String [] str)
        {
                new Demo();
        }
}
```

Output:



Checkbox

- A check box is a graphical component that can be in either an "on" (true) or "off" (false) state.
- Clicking on a check box changes its state from "on" to "off," or from "off" to "on".
- When the user clicks a checkbox, the checkbox states changes and it generates an **action event**.
- The following constructors are used to create Checkbox

S.N.	Constructor & Description
1	**Checkbox()** Creates a check box with an empty string for its label.
2	**Checkbox(String label)** Creates a check box with the specified label.
3	**Checkbox(String label, boolean state)** Creates a check box with the specified label and sets the specified state.
4	**Checkbox(String label, boolean state, CheckboxGroup group)** Constructs a Checkbox with the specified label, set to the specified state, and in the specified check box group.
5	**Checkbox(String label, CheckboxGroup group, boolean state)** Creates a check box with the specified label, in the specified check box group, and set to the specified state.

\

The checkbox class has the following methods.

1. String getLabel(): Returns the label of checkbox.
2. void setLabel(String txt): Sets the label text to txt value in the argument
3. void setCheckboxGroup(CheckboxGroup g): Sets the checkbox group to specified checkbox group g
4. CheckboxGroup getCheckboxGroup(): Returns the checkbox group
5. boolean getState(): Returns true if the checkbox is checked.
6. void setState(boolean st):Sets the state of the checkbox to specified state

CheckboxGroup

- If we want a group of checkboxes in which **only one checkbox at a time can be "on", we** use CheckboxGroup object
- The constructor used to create CheckboxGroup is

public CheckboxGroulp()

Example:

```
import java.awt.*;
class Demo extends Frame
{
        Button btn_OK;
        MenuBar mb;
        Menu mFile,mFont,mColor;
        MenuItem miNew,miClose,miSeparator1;
        MenuItem miFont1,miSeparator2,miFont2,miFont3,miSeparator3;
        MenuItem miRed,miSeparator4,miGreen;
        Panel pTop;
        Panel pMiddle;
        Panel upper;
        Panel pBottom;
        Panel pGender;
        Panel pMarital;
        Label lbl_maritalstatus;
        Label lblcmt;
        Label lblhoby;
        Label lbl_name;
        Label lbl_lastname;
        Label lbl_Gender;
        Choice ch;
        CheckboxGroup cgGender;
        CheckboxGroup cgMarital;

        Checkbox cb_male;
        Checkbox cb_female;
        Checkbox married;
        Checkbox unmarried;
        Checkbox cb_sing,cb_dance,cb_eat,cb_study;
        TextField txtname;
        TextField txtln;
        TextArea txtcmt;
        Font fon=new Font("Arial",Font.BOLD,11);
        Font fon1=new Font("Arial",Font.ITALIC,11);
        Font fon0=new Font("Arial",Font.PLAIN,11);
        //Font fon1=new Font("Arial",Font.ITALIC|Font.BOLD,11);
        public Demo()
        {
                super("This is my first Frame");
                setSize(700,350);
                setLocationRelativeTo(null);
                setResizable(false);
                setBackground(Color.RED);

                setLayout(null);
                setCursor(new Cursor(Cursor.HAND_CURSOR));
                mb=new MenuBar();
                setMenuBar(mb);
                mFile=new Menu("File");
```

```
mFont=new Menu("Font");
mColor=new Menu("Color");
mb.add(mFile);
mb.add(mFont);
mb.add(mColor);
miNew=new  MenuItem("New");
miSeparator1=new  MenuItem("-");
miClose=new MenuItem("Close");
mFile.add(miNew);
mFile.add(miSeparator1);
mFile.add(miClose);
miFont1=new  MenuItem("Arial -Bold");
miSeparator2=new  MenuItem("-");
miFont2=new MenuItem("Arial-Italic");
miFont3=new MenuItem("Arial-Normal");
miSeparator3=new  MenuItem("-");
mFont.add(miFont1);
mFont.add(miSeparator2);
mFont.add(miFont2);
mFont.add(miSeparator3);
mFont.add(miFont3);
miRed=new  MenuItem("Red");
miSeparator4=new  MenuItem("-");
miGreen=new MenuItem("Green");
mColor.add(miRed);
mColor.add(miSeparator4);
mColor.add(miGreen);

pTop=new Panel();
pTop.setLayout(null);
pTop.setBounds(10,50,680,290);
pTop.setBackground(Color.GREEN);

upper =new Panel();
upper.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
upper.setBackground(Color.WHITE);
upper.setBounds(5,5,670,30);
ch=new Choice();
ch.addItem("Choose");
ch.addItem("MR");
ch.addItem("Miss");
ch.addItem("Mrs");

lbl_name=new Label("First Name");
lbl_name.setFont(fon);
txtname=new TextField(27);

lbl_lastname=new Label("Last Name");
lbl_lastname.setFont(fon);
txtln=new TextField(27);

upper.add(ch);
upper.add(lbl_name);
```

```
upper.add(txtname);
upper.add(lbl_lastname);
upper.add(txtln);
pTop.add(upper);

pGender =new Panel();
pGender.setBounds(5,35,670,30);
pGender.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
pGender.setBackground(Color.WHITE);
lbl_Gender=new Label("Gender");

cgGender=new CheckboxGroup();
cgMarital=new CheckboxGroup();

cb_male=new Checkbox("Male",false,cgGender);
cb_male.setEnabled(false);  //disable the checkbox
cb_female=new Checkbox ("Female",false,cgGender);
cb_female.setEnabled(false); //disabling the checkbox

pGender.add(lbl_Gender);
pGender.add(cb_male);
pGender.add(cb_female);
pTop.add(pGender);


pMarital =new Panel();
pMarital.setBounds(5,65,670,30);
pMarital.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
pMarital.setBackground(Color.WHITE);
lbl_maritalstatus=new Label("Marital Status");

married=new Checkbox("Married",false,cgMarital);
married.setEnabled(false); //for disabling the checkbox
unmarried=new Checkbox("Unmmarried",false,cgMarital);
unmarried.setEnabled(false);

pMarital.add(lbl_maritalstatus);
pMarital.add(married);
pMarital.add(unmarried);
pTop.add(pMarital);

pMiddle=new Panel();
pMiddle.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
pMiddle.setBounds(5,90,670,30);
pMiddle.setBackground(Color.RED);
lblhoby=new Label("Hobbies");
lblhoby.setForeground(Color.WHITE);
pMiddle.add(lblhoby);
```

```
                cb_sing=new Checkbox("Singing");
                cb_eat=new Checkbox("Eating");
                cb_dance=new Checkbox("Dancing");
                cb_study=new Checkbox("Studying");

                pMiddle.add(cb_sing);
                pMiddle.add(cb_eat);
                pMiddle.add(cb_dance);
                pMiddle.add(cb_study);
                pTop.add(pMiddle);

                pBottom=new Panel();
                pBottom.setLayout(new FlowLayout(FlowLayout.CENTER,5,5));
                pBottom.setBounds(5,130,670,100);
                pBottom.setBackground(Color.WHITE);
                lblcmt=new Label("Comment");
                pBottom.add(lblcmt);
                txtcmt=new TextArea(4,80);
                pBottom.add(txtcmt);
                pTop.add(pBottom);



                btn_OK=new Button("OK");
                btn_OK.setBounds(300,250,50,20);
                pTop.add(btn_OK);

                add(pTop);
                show();
        }
        public static void main(String [] str)
        {
                new Demo();
        }
}
```

Output: