Department of Software Engineering

# (SW-121: Object Oriented Programming)

**Lab 13:**

**Class:** K20SW (2nd Semester 1st Year)

**Building GUI Components**

**Instructor:** Engr. Abdul Hafeez Babar

# Date: 18 August, 2021

**Time: 12:00 PM – 3:00 PM**

**Mehran University of Engineering and Technology, Shaheed**

**Zulfiqar Ali Bhutto, Campus Khairpur Mir’s**

### Name: Muhammad Saleem \_\_\_\_\_\_\_\_\_\_\_\_\_\_Roll Number: K20SW033

**Score: Signature: Date: 27/11/2021**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Lab performance indicator** | Subject knowledge | Data analysis and interpretation | Ability to conduct experiment | Presentation | Calculation and coding | Observatio n/results | Score |
|  |  |  |  |  |  |  |  |

**Lab Objective:** Upon the successful completion of this practical student will be getting familiar with various layout managers in java.

**Description**

**Layout Managers in Java**

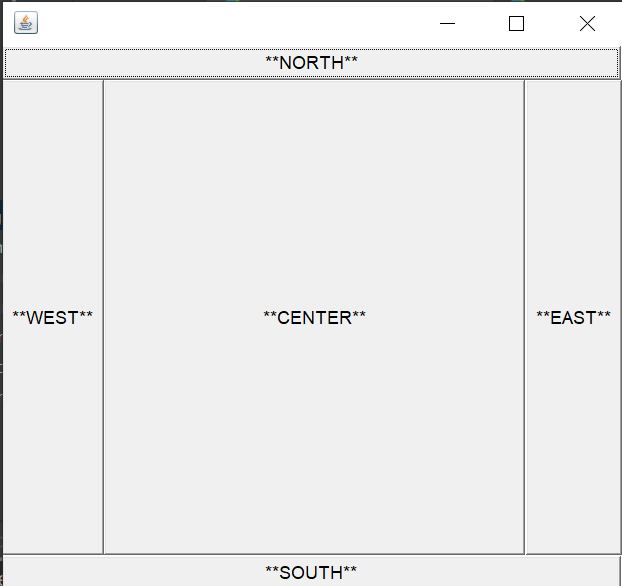
* Layout means the arrangement of components within the container. In other way we can say that placing the components at a particular position within the container. The task of layouting the controls is done automatically by the Layout Manager.
* The layout manager automatically positions all the components within the container. If we do not use layout manager then also the components are positioned by the default layout manager. It is possible to layout the controls by hand but it becomes very difficult because of the following two reasons.
* Java provide us with various layout manager to position the controls. The properties like size,shape and arrangement varies from one layout manager to other layout manager. When the size of the applet or the application window changes the size, shape and arrangement of the components also changes in response i.e. the layout managers adapt to the dimensions of appletviewer or the application window.

### Border Layout:

* BorderLayout is used, when we want to arrange the components in five regions. The five regions can be north, south, east, west and the centre. There are 5 types of constructor in Border Layout. They are as following:
  1. public static final int NORTH
  2. public static final int SOUTH
  3. public static final int EAST
  4. public static final int WEST
  5. public static final int CENTER

### Example:

package Layouts; import java.awt.\*;

public class BorderLayoutProgram { Frame frame; BorderLayoutProgram() {

frame=new Frame();

Button box1=new Button("\*\*NORTH\*\*");; Button box2=new Button("\*\*SOUTH\*\*");; Button box3=new Button("\*\*EAST\*\*");; Button box4=new Button("\*\*WEST\*\*");; Button box5=new Button("\*\*CENTER\*\*");; frame.add(box1, BorderLayout.NORTH); frame.add(box2, BorderLayout.SOUTH); frame.add(box3, BorderLayout.EAST); frame.add(box4, BorderLayout.WEST); frame.add(box5, BorderLayout.CENTER); frame.setSize(400,400); frame.setVisible(true);

}

public static void main(String[] args)

{

new BorderLayoutProgram();

}

}

### Grid Layout:

* Grid Layout is used, when we want to arrange the components in a rectangular grid.

#### There are 3 types of constructor in Grid Layout. They are as following:

1. GridLayout()
2. GridLayout(int rows, int columns)
3. GridLayout(int rows, int columns, inthgap, int vgap)

### Example:

package Layouts; import java.awt.\*; import javax.swing.\*;

public class GridLayoutProgram{ Frame frame1; GridLayoutProgram(){

frame1=new Frame();

Button box1=new Button("\*1\*"); Button box2=new Button("\*2\*");

Button box3=new Button("\*3\*"); Button box4=new Button("\*4\*"); Button box5=new Button("\*5\*"); Button box6=new Button("\*6\*"); Button box7=new Button("\*7\*"); Button box8=new Button("\*8\*"); Button box9=new Button("\*9\*"); frame1.add(box1); frame1.add(box2); frame1.add(box3); frame1.add(box4); frame1.add(box5); frame1.add(box6); frame1.add(box7); frame1.add(box8); frame1.add(box9);

frame1.setLayout(new GridLayout(3,3)); frame1.setSize(500,500); frame1.setVisible(true);

}

public static void main(String[] args) { new GridLayoutProgram();

}

}

### Flow Layout:

* Flow Layout is used, when we want to arrange the components in a sequence one after another.

#### There are 3 types of constructor in the Flow Layout. They are as following:

1. FlowLayout()
2. FlowLayout(int align)
3. FlowLayout(int align, inthgap, intvgap) Example:

Example:

package Layouts; import java.awt.\*;

public class FlowLayoutProgram{ Frame frame1;

FlowLayoutProgram(){ frame1=new Frame();

Button box1=new Button("1"); Button box2=new Button("2"); Button box3=new Button("3"); Button box4=new Button("4"); Button box5=new Button("5"); Button box6=new Button("6"); Button box7=new Button("7"); Button box8=new Button("8"); Button box9=new Button("9"); Button box10=new Button("10");

frame1.add(box1); frame1.add(box2); frame1.add(box3); frame1.add(box4); frame1.add(box5); frame1.add(box6); frame1.add(box7); frame1.add(box8); frame1.add(box9); frame1.add(box10);

frame1.setLayout(new FlowLayout(FlowLayout.LEFT));

frame1.setSize(400,400); frame1.setVisible(true);

}

public static void main(String[] args) { new FlowLayoutProgram();

}

}

### Box Layout:

* Box Layout is used, when we want to arrange the components vertically or horizontally.
* BoxLayout(Container c, int axis)is the only constructor in the Box Layout

### Example:

package Layouts; import javax.swing.\*; import java.awt.\*;

public class BoxLayoutProgram extends Frame { Button buttonBox[];

public BoxLayoutProgram ()

{

buttonBox = new Button [2]; for (int i = 0; i<2; i++)

{

buttonBox[i] = new Button ("\*\* Button " + (i + 1)+" \*\*"); add (buttonBox[i]);

}

setLayout (new BoxLayout (this, BoxLayout.Y\_AXIS)); setSize(500,500);

setVisible(true);

}

public static void main(String args[])

{

BoxLayoutProgram obj=new BoxLayoutProgram();

}

}

### Card Layout

* Card Layout is used, when we want to see only one component at a time.

#### There are 2 types of constructor in the Card Layout. They are as following:

1. CardLayout()
2. CardLayout(inthgap, intvgap)

### Example:

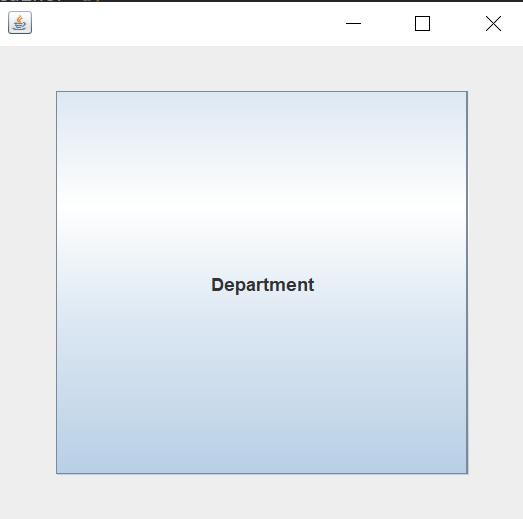
package Layouts; import java.awt.\*;

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

public class CardLayoutProgram extends JFrame implements ActionListener{ CardLayout c\_Card;

JButton box1,box2,box3; Container d; CardLayoutProgram(){

d=getContentPane(); c\_Card=new CardLayout(40,30); d.setLayout(c\_Card);

box1=new JButton("Software"); box2=new JButton("Engineering"); box3=new JButton("Department"); box1.addActionListener(this); box2.addActionListener(this); box3.addActionListener(this);

d.add("P",box1);

d.add("Q",box2);

d.add("R",box3);

}

public void actionPerformed(ActionEvent e)

{

c\_Card.next(d);

}

public static void main(String[] args)

{

CardLayoutProgram obj =new CardLayoutProgram(); obj.setSize(500,500);

obj.setVisible(true); obj.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

}

## Tools/Software Requirement

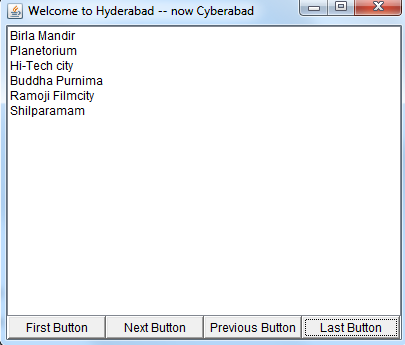
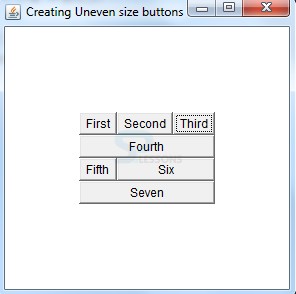
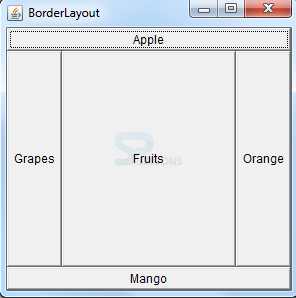
* Windows Operating System
* Java Development Kit (JDK)
* Java Programming Language IDE (Eclipse)

## Helping Material:

* Lecture slides.
* Internet Material:
* <https://www.tutorialspoint.com/awt/awt_borderlayout.htm>
* <https://www.studytonight.com/java/layout-managers.php>
* <https://www.javatpoint.com/java-layout-manager>
* <https://www.brainkart.com/article/Understanding-Layout-Managers---AWT_10666/>

## Lab Tasks

Construct a java program to print out below GUI output.

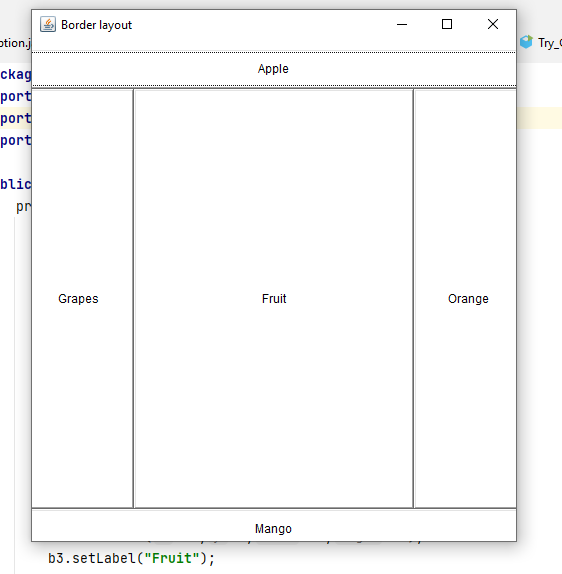


Program 1:

Code:

**package** lab13;  
**import** java.awt.\*;  
**import** java.awt.event.WindowEvent;  
**import** java.awt.event.WindowListener;  
  
**public class** program1 **extends** Frame{  
 program1(){  
 setBounds(420,50,500,540);  
 setTitle(**"Border layout"**);  
 setBackground(Color.***WHITE***);  
 setVisible(**true**);  
 setLayout(**null**);  
 Button b1= **new** Button();  
 b1.setBounds(0,40,500,40);  
 b1.setLabel(**"Apple"**);  
 add(b1);  
 Button b2= **new** Button();  
 b2.setBounds(0,80,110,420);  
 b2.setLabel(**"Grapes"**);  
 add(b2);  
 Button b3= **new** Button();  
 b3.setBounds(110,80,280,420);  
 b3.setLabel(**"Fruit"**);  
 add(b3);  
 Button b4= **new** Button();  
 b4.setBounds(390,80,110,420);  
 b4.setLabel(**"Orange"**);  
 add(b4);  
 Button b5= **new** Button();  
 b5.setBounds(0,500,500,40);  
 b5.setLabel(**"Mango"**);  
 add(b5);  
 addWindowListener(**new** WindowListener(){  
  
 @Override  
 **public void** windowOpened(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowClosing(WindowEvent e) {  
 *// f.dispose();* System.*exit*(0);  
 }  
  
 @Override  
 **public void** windowClosed(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowIconified(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowDeiconified(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowActivated(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowDeactivated(WindowEvent e) {  
  
 }  
 });  
 }  
 **public static void** main(String[] args) {  
 program1 obj1=**new** program1();  
 }  
}

output:

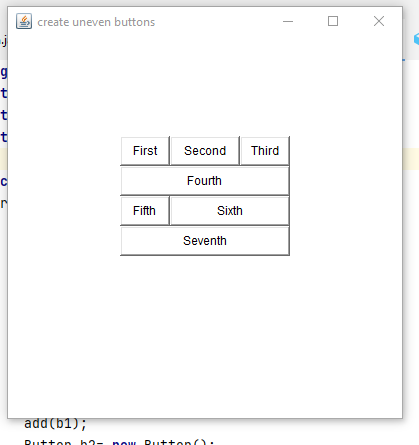


Program 2:

Code:

**package** graphical\_user\_interface;  
**import** java.awt.\*;  
**import** java.awt.event.WindowEvent;  
**import** java.awt.event.WindowListener;  
**public class** Program2 **extends** Frame{  
 Program2(){  
 setBounds(420,50,410,420);  
 setTitle(**"create uneven buttons"**);  
 setBackground(Color.***WHITE***);  
 setVisible(**true**);  
 setLayout(**null**);  
  
 Button b1= **new** Button(**"REGISTER"**);  
 b1.setBounds(120,130,50,30);  
 b1.setLabel(**"First"**);  
 add(b1);  
 Button b2= **new** Button(**"REGISTER"**);  
 b2.setBounds(170,130,70,30);  
 b2.setLabel(**"Second"**);  
 add(b2);  
 Button b3= **new** Button(**"REGISTER"**);  
 b3.setBounds(240,130,50,30);  
 b3.setLabel(**"Third"**);  
 add(b3);  
 Button b4= **new** Button(**"REGISTER"**);  
 b4.setBounds(120,160,170,30);  
 b4.setLabel(**"Fourth"**);  
 add(b4);  
 Button b5= **new** Button(**"REGISTER"**);  
 b5.setBounds(120,190,50,30);  
 b5.setLabel(**"Fifth"**);  
 add(b5);  
 Button b6= **new** Button(**"REGISTER"**);  
 b6.setBounds(170,190,120,30);  
 b6.setLabel(**"Sixth"**);  
 add(b6);  
 Button b7= **new** Button(**"REGISTER"**);  
 b7.setBounds(120,220,170,30);  
 b7.setLabel(**"Seventh"**);  
 add(b7);  
  
 addWindowListener(**new** WindowListener(){  
  
 @Override  
 **public void** windowOpened(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowClosing(WindowEvent e) {  
 *// f.dispose();* System.*exit*(0);  
 }  
  
 @Override  
 **public void** windowClosed(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowIconified(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowDeiconified(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowActivated(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowDeactivated(WindowEvent e) {  
  
 }  
 });  
 }  
  
 **public static void** main(String[] args) {  
 Program2 obj1=**new** Program2();  
 }  
}

output:

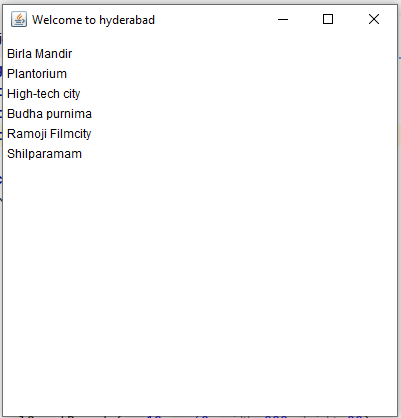


Program 3:

Code:

**package** graphical\_user\_interface;  
**import** java.awt.\*;  
**import** java.awt.event.WindowEvent;  
**import** java.awt.event.WindowListener;  
  
**public class** Program3 **extends** Frame{  
 Program3(){  
 setBounds(420,50,410,420);  
 setTitle(**"Welcome to hyderabad"**);  
 setBackground(Color.***WHITE***);  
 setVisible(**true**);  
 setLayout(**null**);  
 Label l1=**new** Label(**"Birla Mandir"**);  
 l1.setBounds(10,40,200,20);  
 add(l1);  
 Label l2=**new** Label(**"Plantorium"**);  
 l2.setBounds(10,60,200,20);  
 add(l2);  
 Label l3=**new** Label(**"High-tech city"**);  
 l3.setBounds(10,80,200,20);  
 add(l3);  
 Label l4=**new** Label(**"Budha purnima"**);  
 l4.setBounds(10,100,200,20);  
 add(l4);  
 Label l5=**new** Label(**"Ramoji Filmcity"**);  
 l5.setBounds(10,120,200,20);  
 add(l5);  
 Label l6=**new** Label(**"Shilparamam"**);  
 l6.setBounds(10,140,200,20);  
 add(l6);  
 Button b1= **new** Button(**"First Buttton"**);  
 b1.setLabel(**"First Buttton"**);  
 b1.setBounds(0,350,100,30);  
 add(b1);  
 Button b2= **new** Button(**"Next Button"**);  
 b2.setLabel(**"Next Buttton"**);  
 b2.setBounds(100,350,100,30);  
 add(b2);  
 Button b3= **new** Button(**"Previous Button"**);  
 b3.setBounds(200,350,100,30);  
 b3.setLabel(**"Previous Buttton"**);  
 add(b3);  
 Button b4= **new** Button(**"Last Button"**);  
 b4.setBounds(300,350,100,30);  
 b4.setLabel(**"Last Button"**);  
 add(b4);  
 addWindowListener(**new** WindowListener(){  
  
 @Override  
 **public void** windowOpened(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowClosing(WindowEvent e) {  
 *// f.dispose();* System.*exit*(0);  
 }  
  
 @Override  
 **public void** windowClosed(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowIconified(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowDeiconified(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowActivated(WindowEvent e) {  
  
 }  
  
 @Override  
 **public void** windowDeactivated(WindowEvent e) {  
  
 }  
 });  
 }  
  
 **public static void** main(String[] args) {  
 Program3 obj1=**new** Program3();  
  
 }  
}

output:



Program 4:

Code:

## import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.concurrent.BlockingDeque;

class base implements ActionListener {

JFrame f = new JFrame();

JCheckBox adr=new JCheckBox("Android");

JPanel p=new JPanel();

base() {

f.setVisible(true);

f.setLayout(null);

f.setBackground(Color.WHITE);

f.setBounds(100,100,600,450);

f.setResizable(false);

f.getContentPane().setBackground(Color.WHITE);

JCheckBox wind=new JCheckBox("windows");

wind.setBounds(20,30,100,30);

f.add(wind);

adr.setBounds(130,30,100,30);

f.add(adr);

JCheckBox Sol=new JCheckBox("Solaris");

Sol.setBounds(20,80,100,30);

f.add(Sol);

JCheckBox MacOs=new JCheckBox("Mac Os");

MacOs.setBounds(130,80,100,30);

f.add(MacOs);

p.setBounds(20,150,300,250);

p.setBackground(Color.WHITE);

p.setLayout(null);

f.add(p);

p.setEnabled(false);

JLabel l=new JLabel("current state");

l.setBounds(10,10,90,20);

l.setForeground(Color.BLACK);

p.add(l);

JLabel ll=new JLabel("Widows: false");

ll.setBounds(20,40,90,20);

ll.setForeground(Color.BLACK);

p.add(ll);

JLabel lll=new JLabel("Solaris: False");

lll.setBounds(10,70,90,20);

lll.setForeground(Color.BLACK);

p.add(lll);

JLabel l4=new JLabel("Mac: true");

l4.setBounds(10,100,90,20);

l4.setForeground(Color.BLACK);

p.add(l4);

adr.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource()==adr){

}

}

}

public class practice {

public static void main(String[] args) {

base b=new base();

}

}

Output:

