

PROGRAMMING USING JAVA

WEEK 15 ASSIGNMENT

1. What is the use of Layout managers and explain the different types of layout managers with suitable examples.

The Layout managers enable us to control the way in which visual components are arranged in the GUI forms by determining the size and position of components within the containers.

Types of LayoutManager

There are 6 layout managers in Java

- • **FlowLayout:** It arranges the components in a container like the words on a page. It fills the top line from left to right and top to bottom. The components are arranged in the order as they are added i.e. first components appears at top left, if the container is not wide enough to display all the components, it is wrapped around the line. Vertical and horizontal gap between components can be controlled. The components can be left, center or right aligned.
- • **BorderLayout:** It arranges all the components along the edges or the middle of the container i.e. top, bottom, right and left edges of the area. The components added to the top or bottom gets its preferred height, but its width will be the width of the container and also the components added to the left or right gets its preferred width, but its height will be the remaining height of the container. The components added to the center gets neither its preferred height or width. It covers the remaining area of the container.
- • **GridLayout:** It arranges all the components in a grid of equally sized cells, adding them from the left to right and top to bottom. Only one component can be placed in a cell and each region of the grid will have the same size. When the container is resized, all cells are automatically resized. The order of placing the components in a cell is determined as they were added.
- • **GridBagLayout:** It is a powerful layout which arranges all the components in a grid of cells and maintains the aspect ratio of the object whenever the container is resized. In this layout, cells may be different in size. It assigns a consistent horizontal and vertical gap among components. It allows us to specify a default alignment for components within the columns or rows.
- • **BoxLayout:** It arranges multiple components in either vertically or horizontally, but not both. The components are arranged from left to right or top to bottom. If the components are aligned horizontally, the height of all components will be the same and equal to the largest sized components. If the components are aligned vertically, the width of all components will be the same and equal to the largest width components.
- • **CardLayout:** It arranges two or more components having the same size. The components are arranged in a deck, where all the cards of the same size and the only top card are visible at any time. The first component added in the container will be kept at the top of the deck. The default gap at the left, right, top and bottom edges are zero and the card components are displayed either horizontally or vertically.

Example

```
import java.awt.*;

import javax.swing.*;

public class LayoutManagerTest extends JFrame {
    JPanel flowLayoutPanel1, flowLayoutPanel2, gridLayoutPanel1, gridLayoutPanel2,
    gridLayoutPanel3;
    JButton one, two, three, four, five, six;
    JLabel bottom, lbl1, lbl2, lbl3;
    public LayoutManagerTest() {
        setTitle("LayoutManager Test");
        setLayout(new BorderLayout()); // Set BorderLayout for JFrame
        flowLayoutPanel1 = new JPanel();
        one = new JButton("One");
        two = new JButton("Two");
        three = new JButton("Three");
        flowLayoutPanel1.setLayout(new FlowLayout(FlowLayout.CENTER)); // Set FlowLayout Manager
        flowLayoutPanel1.add(one);
        flowLayoutPanel1.add(two);
        flowLayoutPanel1.add(three);
        flowLayoutPanel2 = new JPanel();
        bottom = new JLabel("This is South");
        flowLayoutPanel2.setLayout (new FlowLayout(FlowLayout.CENTER)); // Set FlowLayout Manager
        flowLayoutPanel2.add(bottom);
        gridLayoutPanel1 = new JPanel();
        gridLayoutPanel2 = new JPanel();
        gridLayoutPanel3 = new JPanel();
        lbl1 = new JLabel("One");
        lbl2 = new JLabel("Two");
        lbl3 = new JLabel("Three");
        four = new JButton("Four");
        five = new JButton("Five");
        six = new JButton("Six");
        gridLayoutPanel2.setLayout(new GridLayout(1, 3, 5, 5)); // Set GridLayout Manager
        gridLayoutPanel2.add(lbl1);
        gridLayoutPanel2.add(lbl2);
        gridLayoutPanel2.add(lbl3);
        gridLayoutPanel3.setLayout(new GridLayout(3, 1, 5, 5)); // Set GridLayout Manager
        gridLayoutPanel3.add(four);
        gridLayoutPanel3.add(five);
        gridLayoutPanel3.add(six);
        gridLayoutPanel1.setLayout(new GridLayout(2, 1)); // Set GridLayout Manager
        gridLayoutPanel1.add(gridLayoutPanel2);
        gridLayoutPanel1.add(gridLayoutPanel3);
        add(flowLayoutPanel1, BorderLayout.NORTH);
        add(flowLayoutPanel2, BorderLayout.SOUTH);
        add(gridLayoutPanel1, BorderLayout.CENTER);
        setSize(400, 325);
    }
}
```

```

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setVisible(true);
    }
    public static void main(String args[]) {
        new LayoutManagerTest();
    }
}

```

2. Create a GUI application for student's course registration form with the following fields: Student name, ID number, Age, Gender, course1, course2 (courses should be selected from the list of available courses). Then save all the information.

```

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.Image;
import java.awt.event.*;
import java.awt.event.ActionEvent; import
java.awt.event.ActionListener; import
java.awt.print.*;
import javax.print.Printer;
import java.io.*;
import java.io.IOException;

```

// Creating the fee class

```

public class fee extends Frame {

```

```

    JLabel l1, l2, l3, l4,
        l5, l6, l7, l8,
        l9, l10, l12, l13, l14,
        l11, l15;

```

```

    JTextField tf1, tf2, tf3,
        tf4, tf5, tf6, tf7,
        tf8, tf9, tf10;

```

```

    JTextArea area2, area1;

```

```

    JRadioButton rb1, rb2, rb3,
        rb4, rb5, rb6,
        rb7;

```

```

    JFileChooser f1;

```

```

    // Default constructor to
    // initialize the parameters fee()
    {

```

```
l1 = new JLabel("Fee Report");
l1.setBounds(550, 100, 250, 20);

l2 = new JLabel(
    "Name of the Student:");
l2.setBounds(50, 150, 250, 20);

tf1 = new JTextField();
tf1.setBounds(250, 150, 250, 20);

l3 = new JLabel(
    "Name of the Father:");
l3.setBounds(50, 200, 250, 20);

tf2 = new JTextField();
tf2.setBounds(250, 200, 250, 20);

l4 = new JLabel("Roll Number:");
l4.setBounds(50, 250, 250, 20);

tf3 = new JTextField();
tf3.setBounds(250, 250, 250, 20);

l5 = new JLabel("Email ID:");
l5.setBounds(50, 300, 250, 20);

tf4 = new JTextField();
tf4.setBounds(250, 300, 250, 20);

l6 = new JLabel("Contact Number:");
l6.setBounds(50, 350, 250, 20);

tf5 = new JTextField();
tf5.setBounds(250, 350, 250, 20);

l7 = new JLabel("Address:");
l7.setBounds(50, 400, 250, 20);

area1 = new JTextArea();
area1.setBounds(250, 400, 250, 90);

l9 = new JLabel("Gender:");
l9.setBounds(50, 500, 250, 20);

JRadioButton r5
    = new JRadioButton(" Male");
JRadioButton r6
    = new JRadioButton(" Female");
```

```

r5.setBounds(250, 500, 100, 30);
r6.setBounds(350, 500, 100, 30);

ButtonGroup bg = new ButtonGroup();
bg.add(r5);
bg.add(r6);

l10 = new JLabel("Nationality:");
l10.setBounds(50, 550, 250, 20);

tf6 = new JTextField();
tf6.setBounds(250, 550, 250, 20);

l11 = new JLabel(
    "Year of passing 10th");
l11.setBounds(50, 600, 250, 20);

String language[]
    = { "2016", "2015", "2014" };

final JComboBox cb1
    = new JComboBox(language);

cb1.setBounds(250, 600, 90, 20);

l12 = new JLabel(
    "Year of passing 12th");
l12.setBounds(50, 650, 250, 20);

String languagess[]
    = { "2019", "2018", "2017" };

l13 = new JLabel(
    "Points Secured in 10th:");
l13.setBounds(50, 700, 250, 20);

tf7 = new JTextField();
tf7.setBounds(250, 700, 250, 20);

l14 = new JLabel("Percentage in 12th:");
l14.setBounds(50, 750, 250, 20);

tf8 = new JTextField();
tf8.setBounds(250, 750, 250, 20);

ImageIcon i2 = new ImageIcon("2.png");
JLabel l15
    = new JLabel("", i2, JLabel.CENTER);

```

```

l15.setBounds(900, 50, 600, 200);

final JComboBox cb2
    == new JComboBox(languagess);

cb2.setBounds(250, 650, 90, 20);
l8 = new JLabel(
    "Groups Offered here are:");
l8.setBounds(800, 150, 250, 20);

rb1 = new JRadioButton("SEAS");
rb1.setBounds(550, 150, 100, 30);

rb2 = new JRadioButton("SLABS");
rb2.setBounds(650, 150, 100, 30);

ButtonGroup bg1 = new ButtonGroup();

bg1.add(rb1);
bg1.add(rb2);

rb3 = new JRadioButton("HOSTELLER");
rb3.setBounds(550, 200, 100, 30);

rb4 = new JRadioButton("DAY SCHOLAR");
rb4.setBounds(650, 200, 120, 30);

ButtonGroup bg2 = new ButtonGroup();
bg2.add(rb3);
bg2.add(rb4);

String languages[]
    == { "CSE", "ECE", "EEE",
        "CIVIL", "MECH" };
final JComboBox cb
    == new JComboBox(languages);
cb.setBounds(800, 200, 90, 20);

final JLabel label
    == new JLabel();
label.setBounds(600, 430, 500, 30);
JButton b = new JButton("Show");
b.setBounds(1000, 300, 80, 30);

final DefaultListModel<String> li1
    == new DefaultListModel<>();

```

```

li1.addElement("CSE(2, 50, 000)");
li1.addElement("ECE(2, 50, 000)");
li1.addElement("EEE(2, 50, 000)");
li1.addElement("MECH(2, 50, 000)");
li1.addElement("CIVIL(2, 50, 000)");

final JList<String> list1
    == new JList<>(li1);

list1.setBounds(600, 300, 125, 125);

DefaultListModel<String> li2
    == new DefaultListModel<>();

li2.addElement(
    "2 SHARE(1, 50, 000)");
li2.addElement(
    "3 SHARE(1, 40, 000)");
li2.addElement(
    "5 SHARE(1, 20, 000)");
li2.addElement(
    "8 SHARE(1, 10, 000)");
li2.addElement(
    "bus(40, 000)");

final JList<String> list2
    == new JList<>(li2);
list2.setBounds(
    800, 300, 125, 125);

JButton Receipt
    == new JButton("Generate Receipt");
Receipt.setBounds(600, 490, 150, 30);
JButton b2 = new JButton("Reset");
b2.setBounds(750, 490, 150, 30);
JButton Print = new JButton("Print");
Print.setBounds(900, 490, 150, 30);

area2 = new JTextArea();
area2.setBounds(600, 540, 450, 240);

add(l1);
add(l2);
add(l3);
add(l4);
add(l5);
add(l6);

```

```

add(l7);
add(l8);
add(l9);
add(l10);
add(l11);
add(l12);
add(l13);
add(l14);
add(tf1);
add(tf2);
add(tf3);
add(tf4);
add(tf5);
add(tf6);
add(tf7);
add(tf8);
add(area1);
add(area2);
add(l15);
add(rb1);
add(rb2);
add(rb3);
add(rb4); add(r5);
add(r6); add(cb);

add(cb1); add(cb2);
add(list1);
add(list2); add(b);
add(label);
add(Receipt);
add(b2); add(Print);

```

```

b.addActionListener(new ActionListener() {

    // Method to display the data
    // entered in the text fields
    public void actionPerformed(ActionEvent e)
    {
        String data = "";
        if (list1.getSelectedIndex() != -1) {
            data = "You had selected the Group:"
                ++ list1.getSelectedValue();
            label.setText(data);
        }
        if (list2.getSelectedIndex() != -1) {
            data += " and Hostel with the "
                ++ "facility of: ";

```



```

        for (Object frame :
            list2.getSelectedValues()) { data
            += frame + " ";
        }
    }
    label.setText(data);
}
});

// Reset the text fields
b2.addActionListener(
    new ActionListener() {
        public void actionPerformed(
            ActionEvent e)
        {
            area2.setText("");
            area1.setText(" ");
            tf1.setText("");
            tf2.setText("");
            tf3.setText("");
            tf4.setText("");
            tf5.setText("");
            tf6.setText(" ");
        }
    });

// Implementing the Print action
Print.addActionListener(
    new ActionListener() {
        public void actionPerformed(
            ActionEvent e)
        {
            try {
                area2.print();
            }
            catch (java.awt.print .PrinterException a) {
                System.err.format(    "NoPrinter
                Found", a.getMessage());
            }
        }
    });

// Generating the receipt
Receipt.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        area2.setText(
            "-----"

```

```

+ + "-----FEE RECEIPT----"
+ + "-----"
+ + "-----"
+ + "-----\n");

area2.setText(area2.getText()
+ + "Student Name: "
+ + tf1.getText()
+ + "\n");
area2.setText(area2.getText()
+ + "Father's Name: "
+ + tf2.getText()
+ + "\n");
area2.setText(area2.getText()
+ + "RollNumber: "
+ + tf3.getText()
+ + "\n");
area2.setText(area2.getText()
+ + "Email ID: "
+ + tf4.getText()
+ + "\n");
area2.setText(area2.getText()
+ + "Contact Number: "
+ + tf5.getText()
+ + "\n");
area2.setText(area2.getText()
+ + "Wants to take: "

+ +      cb.getSelectedItem()
      .toString()
+ + "\n");

if (rb1.isSelected()) {
    area2.setText(area2.getText()
+ + "Wants to Join in "
+ + "School of Engineering "
+ + "and Applied Sciences\n");
}
if (rb2.isSelected()) {
    area2.setText(area2.getText()
+ + "Wants to Join in "
+ + "School of Liberal "
+ + "Arts and Sciences\n");
}
if (rb3.isSelected()) {
    area2.setText(area2.getText()
+ + "Wants to be a "
+ + "Hosteller \n");
}

```

```

}
if (rb4.isSelected()) {
    area2.setText(area2.getText()
        + + "Wants to be a "
        + + "Day Scholar \n");
}
area2.setText(area2.getText()
    + + "Had chosen: "
    + + list1.getSelectedValue()
        .toString()
    + + "\n");
area2.setText(area2.getText()
    + + "Had chosen: "
    + + list2.getSelectedValue()
        .toString()
    + + "\n");

int index2 = list2.getSelectedIndex();
if (index2 == 0) {
    area2.setText(area2.getText()
        + + "
        + + "Total amount to be "
        + + "paid is 4 Lakhs \n");
}

if (index2 == 1) {
    area2.setText(area2.getText()
        + + "
        + + "Total amount to be paid "
        + + "is 3.9 Lakhs \n");
}

if (index2 == 2) {
    area2.setText(area2.getText()
        + + "
        + + "Total amount to be paid "
        + + "is 3.8 Lakhs \n");
}

if (index2 == 3) {
    area2.setText(area2.getText()
        + + "
        + + "Total amount to be paid "
        + + "is 3.7 Lakhs \n");
}

if (index2 == 4) {
    area2.setText(area2.getText()

```

```

        ++ "
        ++ "Total amount to be paid "
        ++ "is 2.9 Lakhs \n");
    }

    if (e.getSource() == Receipt) {
        try {
            FileWriter fw
                = new FileWriter( "java.txt",
                true);
            fw.write(area2.getText());
            fw.close();
        }
        catch (Exception ae) {
            System.out.println(ae);
        }
    }

    JOptionPane.showMessageDialog(
        area2, "DATA SAVED SUCCESSFULLY");
};

});
addWindowListener(
    new WindowAdapter() {
        public void windowClosing(
            WindowEvent we)
        {
            System.exit(0); }
    });
setSize(800, 800);
setLayout(null);
setVisible(true);
setBackground(Color.cyan); }
public static void main(String[] args)

{
    new fee();
}
}

```

3. What is an layout manager and types?

A layout manager is an object that implements the `LayoutManager` interface* and determines the size and position of the components within a container

Types of layout manager

- • BorderLayout
- • BoxLayout
- • CardLayout
- • FlowLayout
- • GridBagLayout
- • GridLayout
- • GroupLayout
- • SpringLayout

4. What is the role of layout manager in AWT?

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.

- • It is very tedious to handle a large number of controls within the container.
- • Oftenly the width and height information of a component is not given when we need to arrange them.

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. The layout manager is associated with every Container object. Each layout manager is an object of the class that implements the LayoutManager interface.

Following are the interfaces defining functionalities of Layout Managers.