__/__/____

## * JAVA LAYOUT Managers :-

The LayoutManagers are used to arrange components in a particular manner.

The Java LayoutManagers facilitates us to control the positioning and size of the components in GUI forms.

1. java.awt.BorderLayout
2. java.awt.FlowLayout
3. java.awt.GridLayout

### 1. Java BorderLayout :-

- The BorderLayout is used to arrange the components in five regions : north, south, east, west, and center.

- Each region (area) may contain one component only.

**Constants**

1. BorderLayout.NORTH
2. BorderLayout.SOUTH
3. BorderLayout.EAST

4. BorderLayout.WEST
5. BorderLayout.CENTER

## Constructors

- BorderLayout() :- Creates a border layout but with no gaps between the components.

- BorderLayout(int hgap, int vgap):- Creates a border layout with the given horizontal and vertical gaps between the components.

Example:-

```
import java.awt.*;

public class Border
{
  Border()
{
Frame f = new Frame();

Button b1 = new Button("Notes");
Button b2 = new Button("PPT");
Button b3 = new Button("Programs");
Button b4 = new Button("Blog");
Button b5 = new Button("Projects");
```

__/__/____

```
f.add(b1.BorderLayout.NORTH);
f.add(b2.BorderLayout.SOUTH);
f.add(b3.BorderLayout.EAST);
f.add(b4.BorderLayout.WEST);
f.add(b5.BorderLayout.CENTER);

f.setSize(300,300);
f.setVisible(True);
}
public static void main(String args[])
  {
    Border bo = new Border();
  }
}
```

__/__/____

## ② Java FlowLayout :-

- The Java FlowLayout class is used to arrange the components in a line, one after another (in a flow).
- It is the default layout of the applet.

### Fields of FlowLayout class :-

1. public static final int LEFT
2. public static final int RIGHT
3. public static final int CENTER

### Constructors

1. FlowLayout(): creates a flow layout with centered alignment and a default 5 unit horizontal and vertical gap.
2. FlowLayout(int align) → with Given FlowLayout
3. FlowLayout (int align, int hgap, int vgap)
   - align → given alignment
   - hgap → horizontal gap
   - vgap → Vertical gap

__/__/____

Example :-

```
import java.awt.*;

public class FlowLayoutExample
{
Frame f = new Frame();

FlowLayoutExample()
{
Button b1 = new Button("Notes");
Button b2 = new Button("Blog");
Button b3 = new Button("PPT");
Button b4 = new Button("Programs");
Button b5 = new Button("Instagram");
f.add(b1);
f.add(b2);
f.add(b3);
f.add(b4);
f.add(b5);

f.setLayout(new FlowLayout());
}
public static void main(String args[])
{
FlowLayoutExample obj = new FlowLayoutExample();
}
}
```

__/__/____

## 3 JAVA GridLayout:-

- The JAVA GridLayout class is used to arrange the components in a rectangular grid.
- One component is displayed in each rectangle.

### Constructors

(i) GridLayout() :- It creates a grid layout with one column per component in a row.

(ii) GridLayout(int rows, int columns) :- Creates a grid with the given rows and columns but no gaps between the components.

(iii) GridLayout(int rows, int columns, int hgap, int vgap): with given horizontal and vertical gaps.

### Example :-

```
import java.awt.*;

public class GExample
{
frame f = new frame();
```

```
GExample()
{
    Button b1 = new Button("1");
    Button b2 = new Button("2");
    Button b3 = new Button("3");
    Button b4 = new Button("4");
    Button b5 = new Button("5");

    f.add(b1);
    f.add(b2);
    f.add(b3);
    f.add(b4);
    f.add(b5);

    f.setLayout(new GridLayout());

    f.SetSize(300,300);
    f.setVisible(True);
}

public static void main(String args[])
{
    GExample obj = new GExample();
}
}
```