Practical No 03 - Manual Answer

Advance Java Programming (Government Polytechnic, Pune)

Assignment No 03

Page No 13

ix. Resources used (Additional)

Sr. No.	Name of Resource	Broad Specifications	Quantity	Remarks (if any)
1	Computer System	Lenovo Desktop, Intel Core i3, 2.40 GHz, 4.00 GB RAM	30 (Batch Size)	For all practicals which are Window-based applications
2	Operating System	Windows 8.0, 32-bit OS, x64-based processor		
3	Development Software	JDK 1.8.0_181, Command Prompt, Editors - NetBeans or Eclipse		

Page No 13 and 14

x. Program Code

1. Write java program to demonstrate Grid of 5* 5.

```
import java.awt.Button;
import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

public class GridFiveByFiveDemo extends Frame {
	int i, j;
	GridLayout myLayout;

	public GridFiveByFiveDemo() {
		this.setTitle("Grid 5 x 5 Demo");
		this.setSize(300, 300);

		this.myLayout = new GridLayout();
		this.myLayout.setRows(5);
		this.myLayout.setColumns(5);
		this.setLayout(this.myLayout);

		for(i=1; i<=this.myLayout.getRows(); i++) {
			for(j=1; j<=this.myLayout.getColumns(); j++) {
				this.add(new Button("Button" + i * j));
			}
		}
```

```
            this.addWindowListener(new WindowAdapter() {
                  public void windowClosing(WindowEvent e) {
                        System.exit(0);
                  }
            });

            this.setVisible(true);
      }

      public static void main(String[] args) {
            new GridFiveByFiveDemo();
      }
}
```

2. Write a program to display The Number on Buttons from 0 to 9.

```
import java.applet.Applet;
import java.awt.Button;
import java.awt.Font;
import java.awt.GridLayout;

public class GridLayoutNumbersDemo extends Applet {
      int i, j, k;
      GridLayout myLayout;

      public void init() {
            this.setSize(350, 200);
            this.myLayout = new GridLayout(4, 3);
            this.myLayout.setHgap(5);
            this.myLayout.setVgap(5);
            this.setLayout(this.myLayout);

            this.setFont(new Font("Verdana",
                        Font.BOLD + Font.ITALIC, 20));

            k = 1;
            for(i=1; i<=this.myLayout.getColumns(); i++) {
                  for(j=1; j<=this.myLayout.getColumns(); j++) {
                        this.add(new Button(k + ""));
                        k ++;
                  }
            }
```

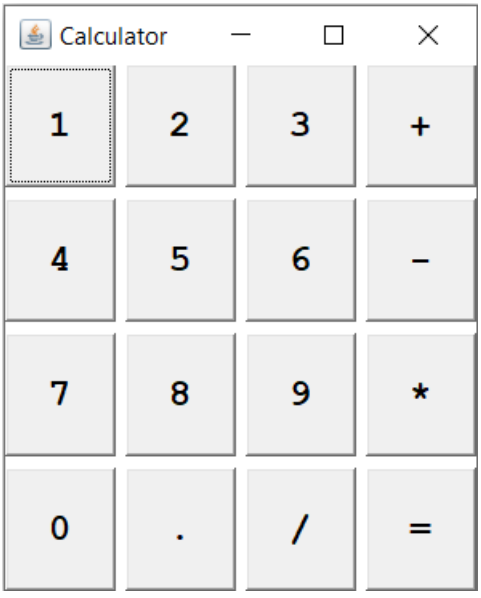
```
            this.add(new Button("0"));
      }
}
```

OR

3. Write the java program to demonstrate the Grid 4 x 4 which displays numbers from 0 to 9 and all arithmetic operators to Design the Calculator.

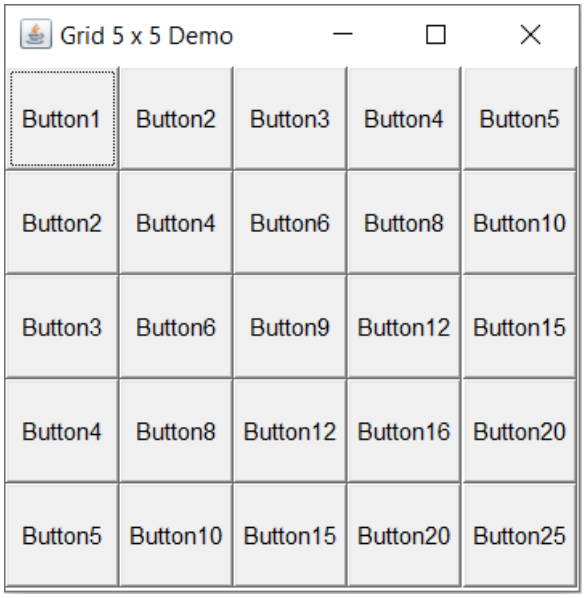
```
import java.awt.Button;
import java.awt.Font;
import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

public class CalculatorDemo extends Frame {
      public CalculatorDemo() {
            this.setTitle("Calculator");
            this.setSize(250, 300);
            this.setLayout(new GridLayout(4, 4, 5, 5));
            this.setFont(new Font("Courier", Font.BOLD, 18));

            Button b1 = new Button("1");
            this.add(b1);
            Button b2 = new Button("2");
            this.add(b2);
            Button b3 = new Button("3");
            this.add(b3);
            Button b4 = new Button("+");
            this.add(b4);

            Button b5 = new Button("4");
            this.add(b5);
            Button b6 = new Button("5");
            this.add(b6);
            Button b7 = new Button("6");
            this.add(b7);
            Button b8 = new Button("-");
            this.add(b8);

            Button b9 = new Button("7");
            this.add(b9);
            Button b10 = new Button("8");
```

```
            this.add(b10);
            Button b11 = new Button("9");
            this.add(b11);
            Button b12 = new Button("*");
            this.add(b12);

            Button b13 = new Button("0");
            this.add(b13);
            Button b14 = new Button(".");
            this.add(b14);
            Button b15 = new Button("/");
            this.add(b15);
            Button b16 = new Button("=");
            this.add(b16);

            this.addWindowListener(new WindowAdapter() {
                  @Override
                  public void windowClosing(WindowEvent e) {
                        System.exit(0);
                  }
            });
            this.setVisible(true);
      }

      public static void main(String[] args) {
            new CalculatorDemo();
      }
}
```

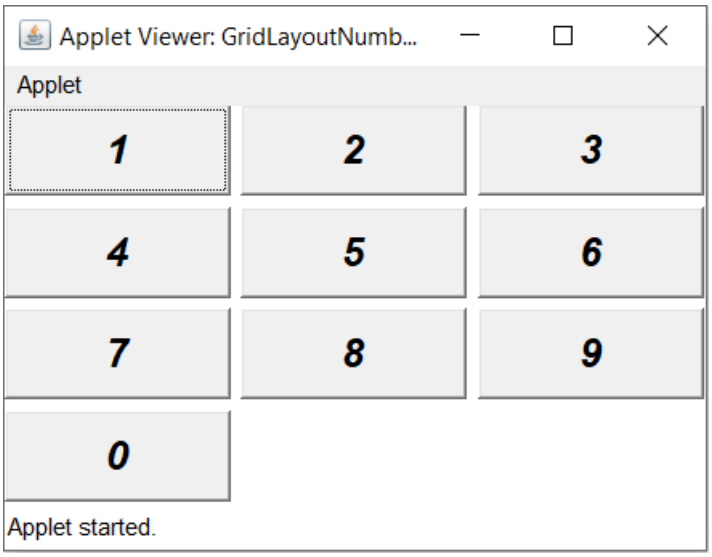


xi. Result (Output of Code)

Output of Program in Q1.



Output of Program in Q2.



Page No 14, 15 and 16

xii. Practical Related Questions

1. Give name of default layout for different containers.

The default layout for Frame and Windows class is BorderLayout. The default layout for Applet and Panel class is FlowLayout.

2. List the names of BorderLayout regions.

a. BorderLayout.NORTH
b. BorderLayout.EAST
c. BorderLayout.SOUTH
d. BorderLayout.WEST
e. BorderLayout.CENTER

3. Write the default horizontal and vertical gap in FlowLayout.

The default horizontal gap and vertical gap in FlowLayout is 5 Units.

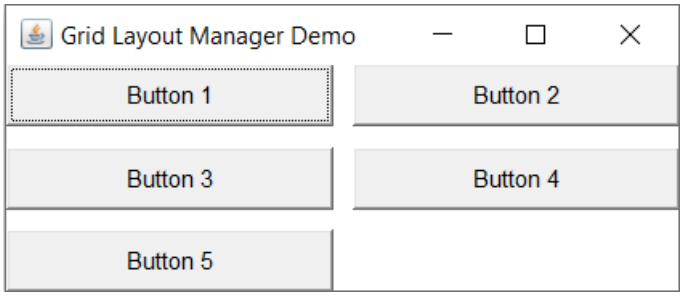
4. Write the use of Insets in border layout.

An Insets object is a representation of the borders of a container. It specifies the space that a container must leave at each of its edges. The space can be a border, a blank space, or a title.

Page No 16 and 17

xiii. Exercise

1. Write a program to generate following output.

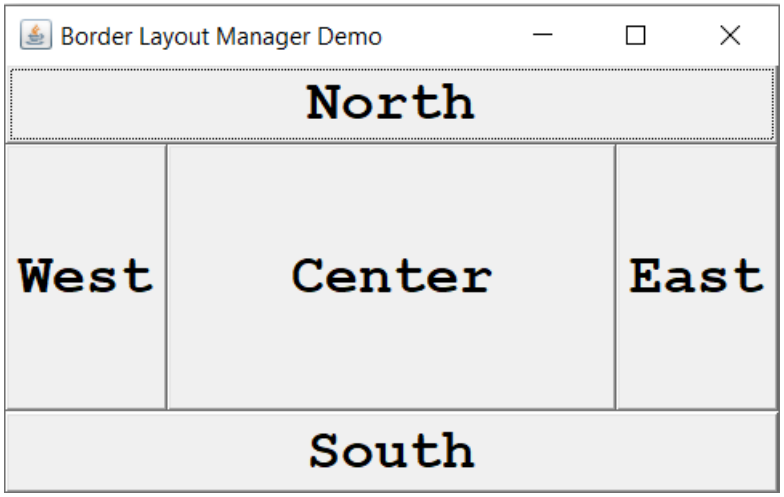


Program:

```
import java.awt.Button;
import java.awt.Frame;
import java.awt.GridLayout;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

public class GridLayoutDemo extends Frame {

      GridLayout myLayout;
```

```
	Button Button1, Button2, Button3, Button4, Button5;

	public GridLayoutDemo() {
		this.setTitle("Grid Layout Manager Demo");
		this.setSize(350, 150);

		this.myLayout = new GridLayout(3, 2, 10, 10);
		this.setLayout(this.myLayout);

		this.Button1 = new Button("Button 1");
		this.add(this.Button1);

		this.Button2 = new Button("Button 2");
		this.add(this.Button2);

		this.Button3 = new Button("Button 3");
		this.add(this.Button3);

		this.Button4 = new Button("Button 4");
		this.add(this.Button4);

		this.Button5 = new Button("Button 5");
		this.add(this.Button5);

		this.addWindowListener(new WindowAdapter() {
			@Override
			public void windowClosing(WindowEvent e) {
				System.exit(0);
			}
		});

		this.setVisible(true);
	}

	public static void main(String[] args) {
		new GridLayoutDemo();
	}
}
```

2. Write a program to generate following output using BorderLayout.



Program:

```
import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.Font;
import java.awt.Frame;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

public class BorderLayoutDemo extends Frame {

	Button btnNorth, btnSouth, btnCenter, btnEast, btnWest;

	public BorderLayoutDemo() {
		this.setTitle("Border Layout Manager Demo");
		this.setSize(400, 250);
		this.setLayout(new BorderLayout());
		this.setFont(
				new Font("Courier", Font.BOLD, 28));

		this.btnNorth = new Button("North");
		this.add(this.btnNorth, BorderLayout.NORTH);

		this.btnEast = new Button("East");
		this.add(this.btnEast, BorderLayout.EAST);

		this.btnSouth = new Button("South");
		this.add(this.btnSouth, BorderLayout.SOUTH);
```

```
		this.btnWest = new Button("West");
		this.add(this.btnWest, BorderLayout.WEST);

		this.btnCenter = new Button("Center");
		this.add(this.btnCenter, BorderLayout.CENTER);

		this.addWindowListener(new WindowAdapter() {
			@Override
			public void windowClosing(WindowEvent e) {
				System.exit(0);
			}
		});

		this.setVisible(true);
	}

	public static void main(String[] args) {
		new BorderLayoutDemo();
	}
}
```