MODULE – 4

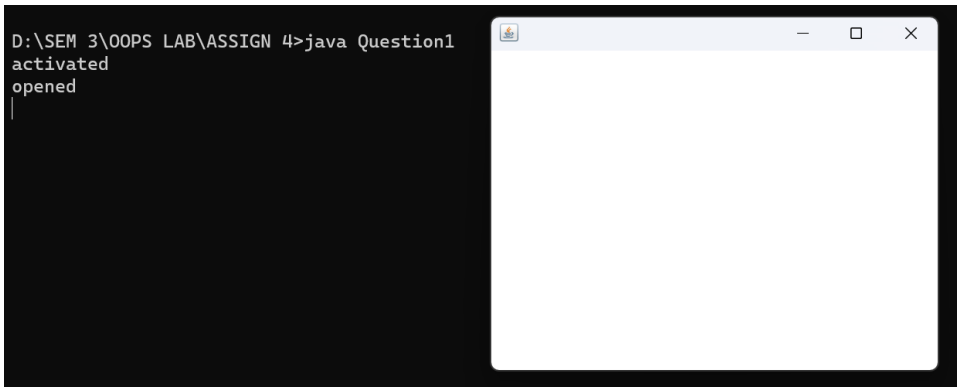
1. Write a program to demonstrate different Window handling events.

CODE:

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

public class Question1 extends Frame implements WindowListener {
   Question1() {
      addWindowListener(this);
      setSize (400, 400);
      setLayout (null);
      setVisible (true);
   }

   public static void main(String[] args) {
      new Question1();
   }

   public void windowActivated (WindowEvent arg0) {
      System.out.println("activated");
   }

   public void windowClosed (WindowEvent arg0) {
      System.out.println("closed");
   }

   public void windowClosing (WindowEvent arg0) {
      System.out.println("closing");
      dispose();
```

```
    }

    public void windowDeactivated (WindowEvent arg0) {
        System.out.println("deactivated");
    }

    public void windowDeiconified (WindowEvent arg0) {
        System.out.println("deiconified");
    }

    public void windowIconified(WindowEvent arg0) {
        System.out.println("iconified");
    }

    public void windowOpened(WindowEvent arg0) {
        System.out.println("opened");
    }
}
```

OUTPUT:



activated

opened

closing

deactivated

closed

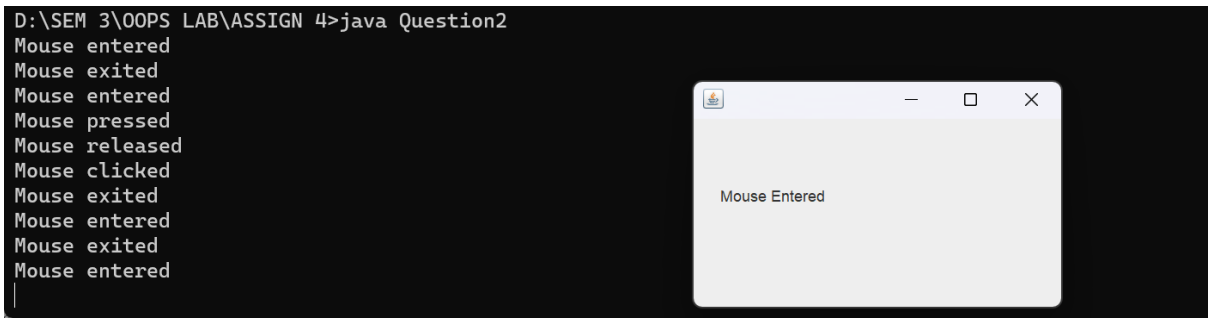
2. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().

CODE:

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


public class Question2 extends JFrame implements MouseListener{
   Label l;
   Question2(){
      addMouseListener(this);
            setDefaultCloseOperation(EXIT_ON_CLOSE);
      l=new Label();
      l.setBounds(20,50,100,20);
      add(l);
      setSize(300,300);
      setLayout(null);
      setVisible(true);
   }
public static void main(String[] args) {
   new Question2();
}
   public void mouseClicked(MouseEvent e) {
      l.setText("Mouse Clicked");
           System.out.println("Mouse clicked");
   }
   public void mouseEntered(MouseEvent e) {
      l.setText("Mouse Entered");
            System.out.println("Mouse entered");
   }
   public void mouseExited(MouseEvent e) {
      l.setText("Mouse Exited");
           System.out.println("Mouse exited");
```

```
    }
    public void mousePressed(MouseEvent e) {
        l.setText("Mouse Pressed");
            System.out.println("Mouse pressed");
    }
    public void mouseReleased(MouseEvent e) {
        l.setText("Mouse Released");
            System.out.println("Mouse released");
    }

}
```

OUTPUT:



Mouse entered

Mouse exited

Mouse entered

Mouse pressed

Mouse released

Mouse clicked

Mouse exited

Mouse entered

Mouse exited

Mouse entered

Mouse exited

Mouse entered

Mouse exited

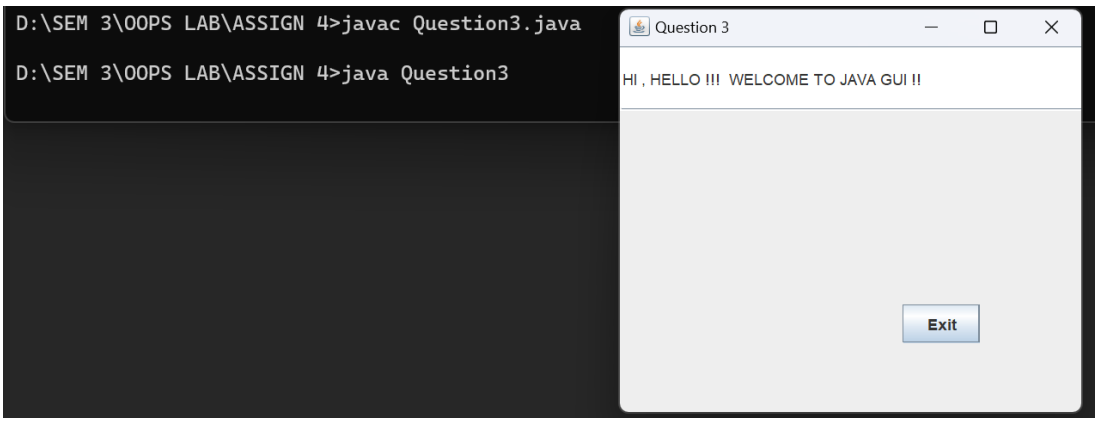
3. Write a program to demonstrate different keyboard handling events.

CODE:

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

class Question3 implements KeyListener, ActionListener {
   static JFrame frame;
   static JTextField input, output;

   public static void main(String[] args) {
      frame = new JFrame("Question 3");
      frame.setSize(500, 500);
      frame.setLayout(null);
      output = new JTextField();
      output.setBounds(0, 0, 500, 50);
      frame.add(output);
      input = new JTextField();
      input.setBounds(0, 400, 500, 50);
      frame.add(input);
      JButton exit = new JButton("Exit");
      exit.setBounds(220, 200, 60, 30);
      frame.add(exit);
      Question3 obj = new Question3();
      input.addKeyListener(obj);
      exit.addActionListener(obj);
      frame.setVisible(true);
   }

   public void actionPerformed(ActionEvent ae) {
      frame.dispose();
```

```
    }

    public void keyReleased(KeyEvent e) {
        output.setText("");
        output.setText("Key Released : "+e.getKeyCode());
        if(Character.isLetter(e.getKeyChar()))
            keyTyped(e);
        if(Character.isDigit(e.getKeyChar()))
            keyTyped(e);
    }

    public void keyPressed(KeyEvent e) {
        output.setText("");
        output.setText("Key Pressed : "+e.getKeyCode());
        if(Character.isLetter(e.getKeyChar()))
            keyTyped(e);
        if(Character.isDigit(e.getKeyChar()))
            keyTyped(e);
    }

    public void keyTyped(KeyEvent e) {
        output.setText("");
        output.setText("Key Typed : "+e.getKeyChar());
    }
}
```

OUTPUT:

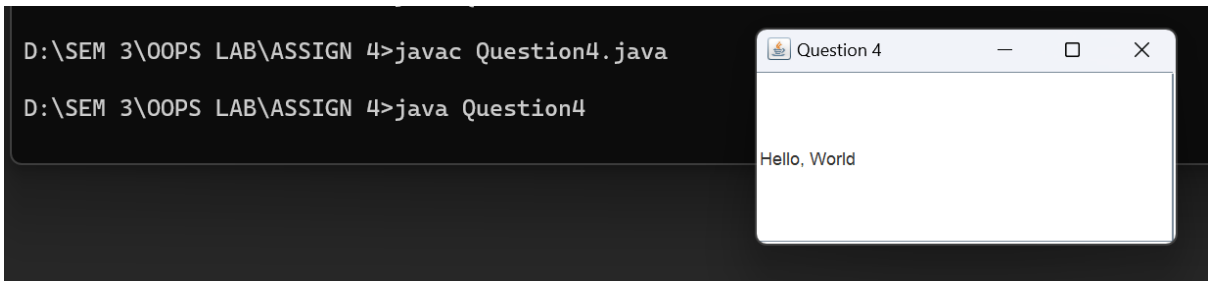


4. Write a program to generate a window without an applet window using main() function.

CODE:

```
import javax.swing.*;


class Question4 {
   public static void main(String[] args) {
      JFrame frame = new JFrame("Question 4");
      frame.setSize(300, 300);
      frame.add(new JTextField("Hello, World"));
      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }
}
```

OUTPUT:



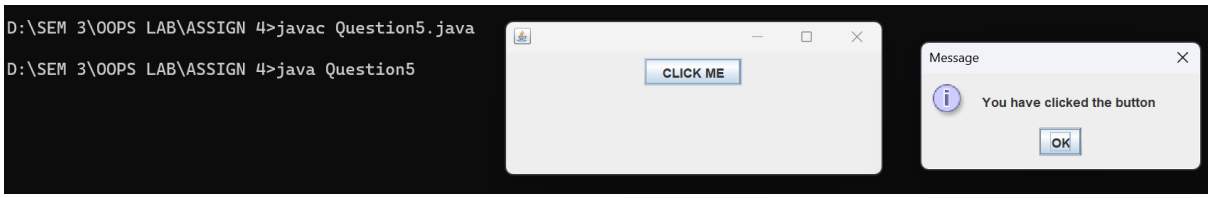
5. Write a program to demonstrate the use of push buttons.

CODE:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;


class Question5 {
   public static void main(String[] args) {
      new Question5();
   }
```

```
    ActionListener e=new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JOptionPane.showMessageDialog(null,"You have clicked the button");
        }
    };
    public Question5() {
        JFrame f = new JFrame();
        JButton b1 = new JButton();
        b1.setText("CLICK ME");
        b1.addActionListener(e);
        f.add(b1);
        f.setLayout(new FlowLayout());
        f.setVisible(true);
        f.setSize(400, 400);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

OUTPUT:



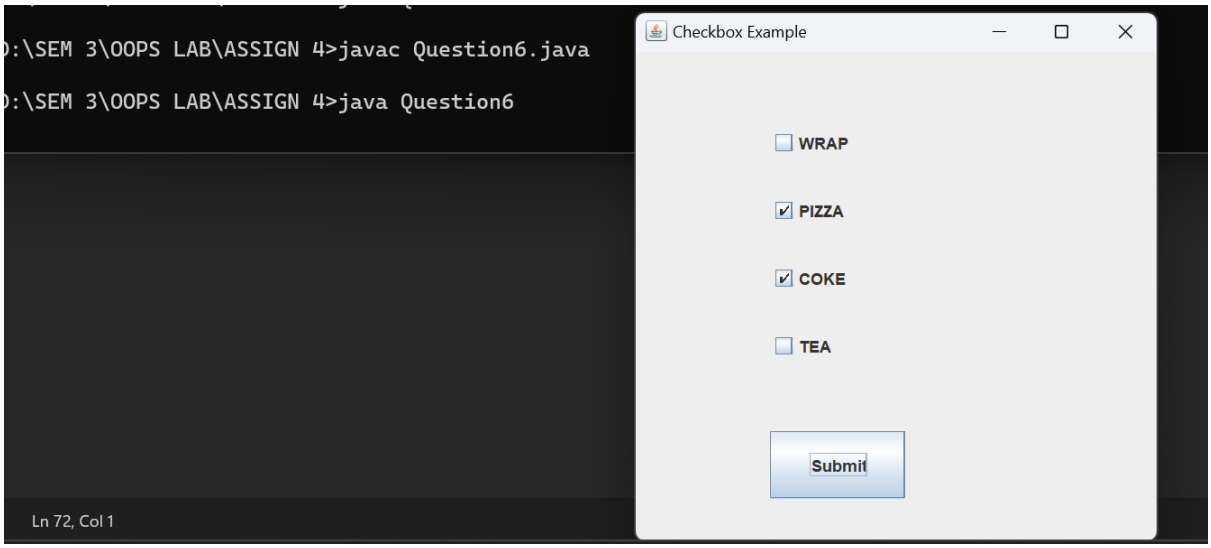
6. WAP to create a Menu using the frame.

CODE:

```
import javax.swing.*;
import java.util.*;
import java.awt.*;

public class Question6 {
```

```
    public Question6() {
        JFrame f = new JFrame("Checkbox Example");
        String[] langs = {"WRAP", "PIZZA", "COKE", "TEA"};

        int xDisp = 100, yDisp = 30, width = 75, height = 75;
        ArrayList<JCheckBox> checkBoxes = new ArrayList<JCheckBox>();
        for(int i = 0; i < langs.length; i++) {
            checkBoxes.add(new JCheckBox(langs[i], false));
            checkBoxes.get(i).setBounds(xDisp, yDisp, width, height);
            yDisp += 50;
        }

        for(JCheckBox checkBox: checkBoxes)
            f.add(checkBox);
        JButton btn = new JButton("Submit");
        btn.setBounds(xDisp, yDisp + 50, 100, 50);

        f.add(btn);

        f.setLayout(null);
        f.setSize(400, 400);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {
        new Question6();
    }
}
```

OUTPUT:



7. WAP to create a Frame that display the student information.

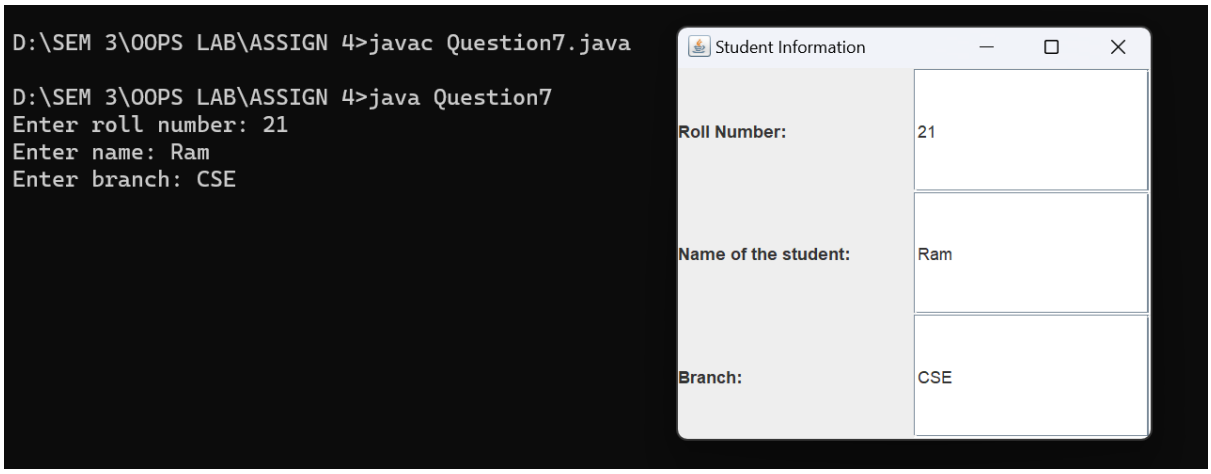
CODE:

```
import javax.swing.*;
import java.awt.*;
import java.util.*;

class Student {
   JLabel L1, L2, L3;
   JTextField tf1, tf2, tf3;

   public Student() { initGui(); }

   public void initGui() {
      Scanner sc = new Scanner(System.in);

      JFrame frame = new JFrame("Student Information");

      this.L1 = new JLabel("Roll Number: ");
      this.L2 = new JLabel("Name of the student: ");
      this.L3 = new JLabel("Branch: ");
```

```
    this.tf1 = new JTextField(20);
    this.tf2 = new JTextField(20);
    this.tf3 = new JTextField(20);

    System.out.print("Enter roll number: ");
    String rollNumber = sc.nextLine();
    System.out.print("Enter name: ");
    String name = sc.nextLine();
    System.out.print("Enter branch: ");
    String branch = sc.nextLine();

    this.tf1.setText(rollNumber);
    this.tf2.setText(name);
    this.tf3.setText(branch);

    Container container = frame.getContentPane();
    container.setLayout(new GridLayout(3, 2));
     // container.setLayout(null);

    container.add(this.L1);
    container.add(this.tf1);
    container.add(this.L2);
    container.add(this.tf2);
    container.add(this.L3);
    container.add(this.tf3);

    frame.setSize(350, 300);
    frame.setVisible(true);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

```
}

public class Question7 {
    public static void main(String[] args) {
        new Student();
    }
}
```

OUTPUT:



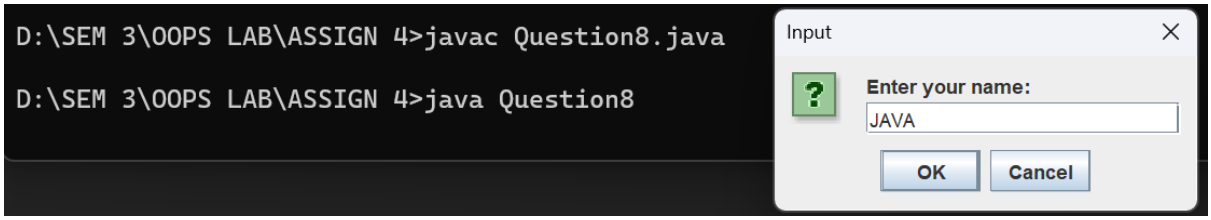
8. WAP to create a Dialogbox.

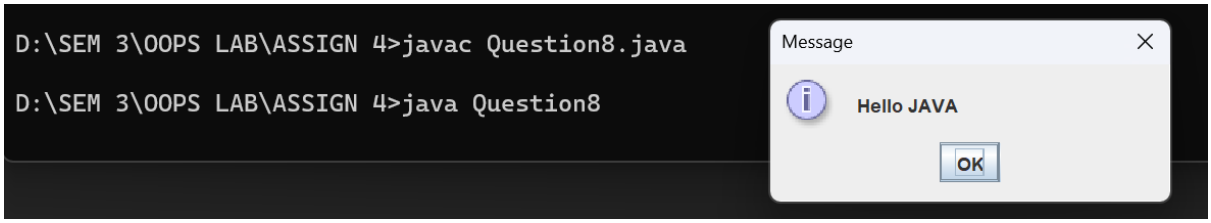
CODE:

```
import javax.swing.*;

public class Question8 {
    Question8() {
        JFrame frame = new JFrame();
        String name = JOptionPane.showInputDialog(frame, "Enter your name: ");
        JOptionPane.showMessageDialog(frame, "Hello " + name);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    public static void main(String[] args) {
        new Question8();
        System.exit(0);
```

```
    }
}
```

OUTPUT:





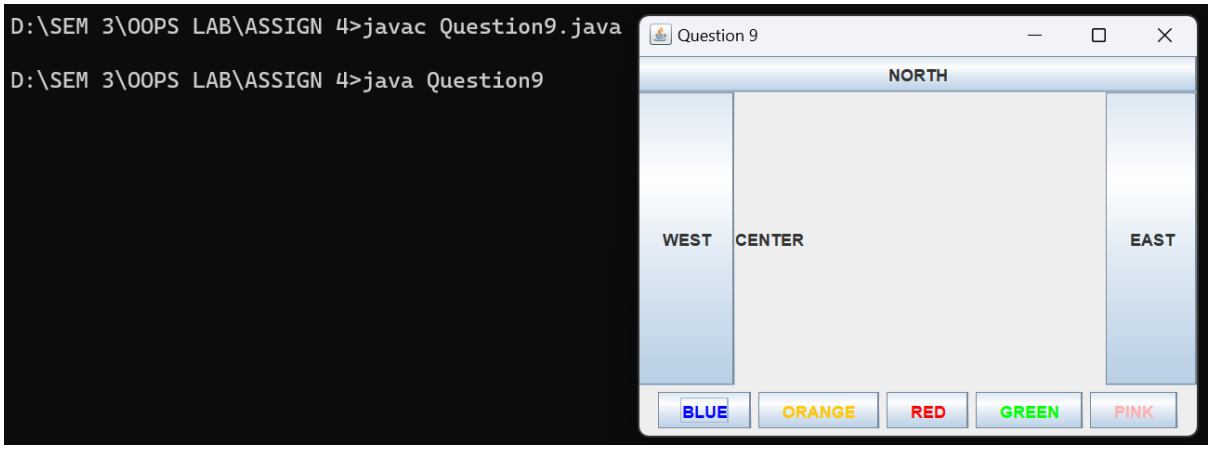
9. WAP to implement the FlowLayout and BorderLayout.

CODE:

```
import javax.swing.*;
import java.awt.*;

class Question9 {
    private static void buildButton(String value, Color color, JPanel toAdd) {
        JButton button = new JButton(value);
        button.setForeground(color);
        toAdd.add(button);
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Question 9");
        frame.setSize(600, 600);

        JPanel buttonPanel = new JPanel(new FlowLayout());
        buildButton("BLUE", Color.BLUE, buttonPanel);
        buildButton("ORANGE", Color.ORANGE, buttonPanel);
```

```
        buildButton("RED", Color.RED, buttonPanel);
        buildButton("GREEN", Color.GREEN, buttonPanel);
        buildButton("PINK", Color.PINK, buttonPanel);

        JPanel mainPanel = new JPanel(new BorderLayout());
        mainPanel.add(buttonPanel, BorderLayout.SOUTH);
        mainPanel.add(new JButton("NORTH"), BorderLayout.NORTH);
        mainPanel.add(new JButton("WEST"), BorderLayout.WEST);
        mainPanel.add(new JButton("EAST"), BorderLayout.EAST);
        mainPanel.add(new JLabel("CENTER"), BorderLayout.CENTER);

        frame.setContentPane(mainPanel);

        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

OUTPUT:



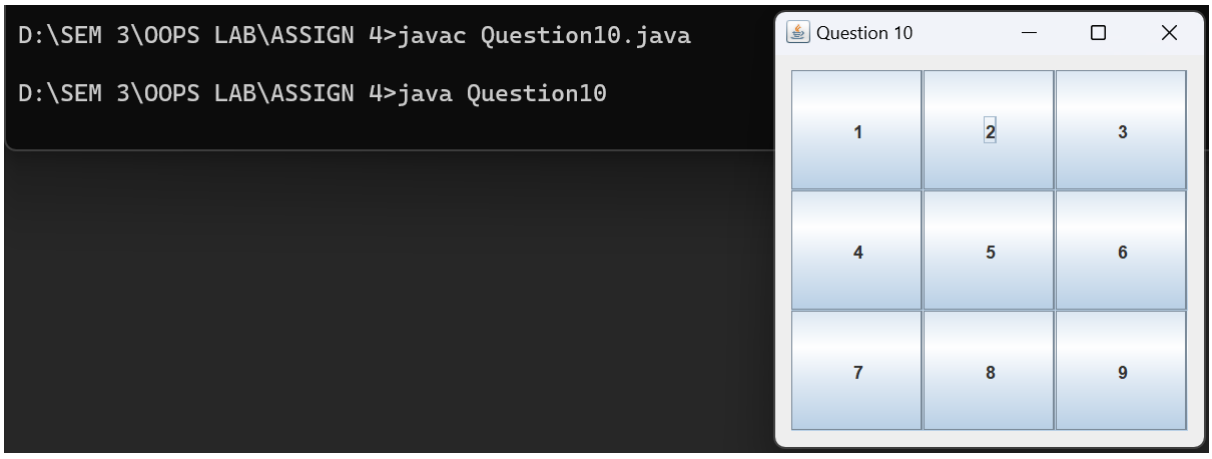
10. WAP to implement the GridLayout and CardLayout.

CODE:

```
import javax.swing.*;
import java.awt.*;
```

```
class Question10 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Question 10");
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(3, 3));
        JButton button;
        for(int i = 1; i < 10; i++) {
            button = new JButton(i+"");
            panel.add(button);
        }

        JPanel main = new JPanel();
        main.setLayout(new CardLayout(10, 10));
        main.add("Numbers", panel);

        frame.setContentPane(main);

        frame.setSize(300, 300);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

OUTPUT:



11. WAP to implement the GroupLayout and BoxLayout.

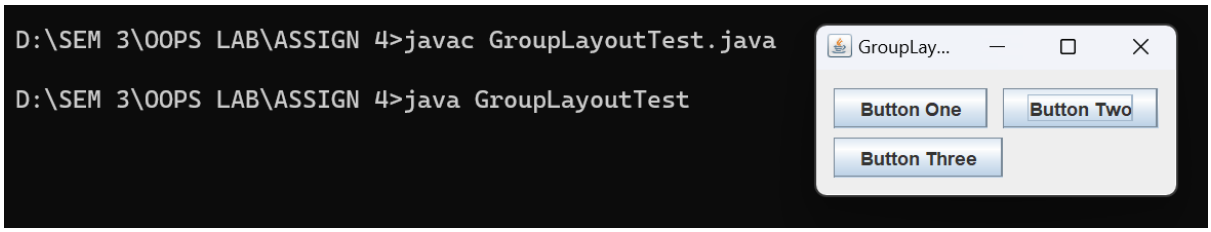
#GROUPLAYOUT:

CODE:

```
import javax.swing.*;
import java.awt.*;
import static javax.swing.GroupLayout.Alignment.*;

class GroupLayoutTest {
   public static void main(String[] args) {
      JFrame frame = new JFrame("GroupLayout Implementation");
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      Container myPanel = frame.getContentPane();

      GroupLayout groupLayout = new GroupLayout(myPanel);
      groupLayout.setAutoCreateGaps(true);
      groupLayout.setAutoCreateContainerGaps(true);
      myPanel.setLayout(groupLayout);

      JButton b1 = new JButton("Button One");
      JButton b2 = new JButton("Button Two");
      JButton b3 = new JButton("Button Three");

      groupLayout.setHorizontalGroup(groupLayout.createSequentialGroup()

.addGroup(groupLayout.createParallelGroup(LEADING).addComponent(b1).addComponent(b3))
            .addGroup(groupLayout.createParallelGroup(TRAILING).addComponent(b2)));

      groupLayout.setVerticalGroup(groupLayout.createSequentialGroup()

.addGroup(groupLayout.createParallelGroup(BASELINE).addComponent(b1).addComponent(b2))
            .addGroup(groupLayout.createParallelGroup(BASELINE).addComponent(b3)));
```

```
        frame.pack();
        frame.setVisible(true);
    }
}
```

OUTPUT:



 # <u>BOXLAYOUT:</u>

CODE:

```
import javax.swing.*;
import java.awt.*;

class BoxLayoutTest{
    public static void main(String[] args) {
        JFrame frame = new JFrame("Box Layout Implementation");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JButton jb1 = new JButton("Button 1 -");
        JButton jb2 = new JButton("Button 2 --------");
        JButton jb3 = new JButton("Button 3 -");
        JButton jb4 = new JButton("Button 4 --------");
        JButton jb5 = new JButton("Button 5 -");
        JButton jb6 = new JButton("Button 6 --------");

        JPanel panel1 = new JPanel();
        JPanel panel2 = new JPanel();
        JPanel panel3 = new JPanel();
```

```
panel1.setBorder(BorderFactory.createTitledBorder("LEFT"));
panel2.setBorder(BorderFactory.createTitledBorder("CENTER"));
panel3.setBorder(BorderFactory.createTitledBorder("RIGHT"));

BoxLayout layout1 = new BoxLayout(panel1, BoxLayout.Y_AXIS);
BoxLayout layout2 = new BoxLayout(panel2, BoxLayout.Y_AXIS);
BoxLayout layout3 = new BoxLayout(panel3, BoxLayout.Y_AXIS);
panel1.setLayout(layout1);
panel2.setLayout(layout2);
panel3.setLayout(layout3);

jb1.setAlignmentX(Component.LEFT_ALIGNMENT);
jb2.setAlignmentX(Component.LEFT_ALIGNMENT);
panel1.add(jb1);
panel1.add(jb2);

jb3.setAlignmentX(Component.CENTER_ALIGNMENT);
jb4.setAlignmentX(Component.CENTER_ALIGNMENT);
panel2.add(jb3);
panel2.add(jb4);

jb5.setAlignmentX(Component.RIGHT_ALIGNMENT);
jb6.setAlignmentX(Component.RIGHT_ALIGNMENT);
panel3.add(jb5);
panel3.add(jb6);

frame.setLayout(new FlowLayout());
frame.add(panel1);
frame.add(panel2);
frame.add(panel3);
```

```
        frame.pack();
        frame.setVisible(true);
    }
}
```

OUTPUT: