Q.Write a program to show different components and containers in swing.

```
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JScrollBar;
import javax.swing.JScrollPane;
import javax.swing.JSlider;
import javax.swing.JTabbedPane;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import java.awt.FlowLayout;
import java.awt.GridLayout;


public class ComponentDemo {
   public static void main(String[] args) {
       JFrame jFrame = new JFrame();
       jFrame.setSize(600, 400);
       jFrame.setLocation(300, 200);
       jFrame.setTitle("JFrame Demo");

       jFrame.setLayout(new GridLayout(5,5));
       // JPanel jPanel = new JPanel();
       JLabel jLabel = new JLabel("This is new display",
SwingConstants.CENTER);
       jFrame.add(jLabel);

       JTextArea jTextArea = new JTextArea("This is text area");
       JScrollPane jScrollPane = new JScrollPane(jTextArea);
```

```
        jFrame.add(jScrollPane);

        JButton jButton = new JButton("Click here");
        jButton.setSize(20, 40);
        jFrame.add(jButton);

        JCheckBox checkBox = new JCheckBox("Check box 1");
        jFrame.add(checkBox);

        JPanel jPanel = new JPanel();
        JRadioButton jRadioButton1 = new JRadioButton("Radio 1");
        JRadioButton jRadioButton2 = new JRadioButton("Radio 2");
        ButtonGroup buttonGroup = new ButtonGroup();
        buttonGroup.add(jRadioButton1);
        buttonGroup.add(jRadioButton2);
        jPanel.add(jRadioButton1);
        jPanel.add(jRadioButton2);
        jFrame.add(jPanel);

        String[] itemList = new String[] {"Item 1", "Item 2", "Item 3"};
        JList<String> jList = new JList<>(itemList);
        jFrame.add(jList);

        String[] comboBoxList = new String[] {"Item 1", "Item 2", "Item 3"};
        JComboBox<String> jComboBox = new JComboBox<>(comboBoxList);
        jFrame.add(jComboBox);

        JSlider slider = new JSlider(0, 100, 40);
        jFrame.add(slider);


        String[][] tableData = new String[][] {{"1", "2", "3"}, {"4", "5",
"6"}};
        String[] tableHeading = new String[] {"Heading 1", "Heading 2", "Heading
3"};
        JTable table = new JTable(tableData, tableHeading);
        jFrame.add(table);
        JTabbedPane jTabbedPane = new JTabbedPane();
        jTabbedPane.addTab("Tab 1", new JLabel("Label 1"));
        jTabbedPane.addTab("Tab 2", new JLabel("Label 2"));
        jFrame.add(jTabbedPane);

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

Q.Write a program to demonstrate the Flow Layout in swing.

```
import javax.swing.JButton;
import javax.swing.JFrame;
import java.awt.FlowLayout;

public class FlowLayoutExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("FlowLayout Demo");
        frame.setLayout(new FlowLayout()); // default is CENTER alignment

        frame.add(new JButton("One"));
        frame.add(new JButton("Two"));
        frame.add(new JButton("Three"));
        frame.add(new JButton("Four"));
        frame.add(new JButton("Five"));

        frame.setSize(400, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Q. Write a program to demonstrate the Grid Layout in swing.

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

public class GridLayoutExample {
   public static void main(String[] args) {
       JFrame frame = new JFrame("GridLayout Demo");
       frame.setLayout(new GridLayout(3, 3));

       for (int i = 1; i <= 5; i++) {
           frame.add(new JButton("Button " + i));
       }

       frame.setSize(400, 200);
       frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       frame.setVisible(true);
   }
}
```

Q.Write a program to handle action events in a Swing component using an ActionListener.

```java
package swing.project.eventhandling;

import javax.swing.JButton;
import javax.swing.JFrame;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ActionEventListenerExample {
   public static void main(String[] args) {
       JFrame jFrame = new JFrame();
       jFrame.setSize(600, 400);
       jFrame.setLocation(300, 200);
       jFrame.setTitle("Action Event Demo");

       JButton jButton = new JButton("Button");
       jFrame.add(jButton);

       jButton.addActionListener(new ActionListener() {

           @Override
           public void actionPerformed(ActionEvent actionEvent) {
              System.out.println("Button Clicked");
           }
       });

       jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       jFrame.setVisible(true);
   }


   public static class MyActionListener implements ActionListener {

       @Override
       public void actionPerformed(ActionEvent e) {
           System.out.println("Button Clicked");
       }

   }
}
```

Q.Write a program to handle item events in a Swing component using an ItemEvent listener.

```
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;

import javax.swing.JCheckBox;
import javax.swing.JFrame;
```

```
public class ItemListenerExample {
    public static void main(String[] args) {
       JFrame jFrame = new JFrame();
       jFrame.setSize(600, 400);
       jFrame.setLocation(300, 200);
       jFrame.setTitle("Action Event Demo");

       JCheckBox checkBox = new JCheckBox("Accept Terms");
       jFrame.add(checkBox);

       checkBox.addItemListener(new ItemListener() {

           @Override
           public void itemStateChanged(ItemEvent e) {
               if (e.getStateChange() == ItemEvent.SELECTED) {
                   System.out.println("Checkbox selected");
               } else {
                   System.out.println("Checkbox deselected");
               }
           }

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

Q. Write a program to handle mouse events in a Swing component using a mouse event listener.

```
import java.awt.FlowLayout;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;

import javax.swing.JFrame;
import javax.swing.JLabel;

public class MouseEventListenerExample {
   public static void main(String[] args) {
       JFrame frame = new JFrame("Mouse Listener Example");
       JLabel label = new JLabel("Click inside the frame");
       frame.add(label);

       frame.addMouseListener(new MouseListener() {

           @Override
           public void mouseClicked(MouseEvent e) {
               label.setText("Clicked at: " + e.getX() + ", " + e.getY());
           }

           @Override
           public void mousePressed(MouseEvent e) {
            System.out.println(“Mouse is pressed”);
           }

           @Override
           public void mouseReleased(MouseEvent e) {
              System.out.println(“Mouse is released”);
           }

           @Override
           public void mouseEntered(MouseEvent e) {
               System.out.println(“Mouse is entered”);
           }

           @Override
           public void mouseExited(MouseEvent e) {
              System.out.println(“Mouse is existed”);
           }

       });

       frame.setSize(300, 200);
       frame.setLayout(new FlowLayout());
       frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       frame.setVisible(true);
   }
}
```

Write a simple program that takes two numbers as input using JTextField, and includes an 'Add' button. When the button is clicked, it should display the result.

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ClaculatorDemo {
   public static void main(String[] args) {
       JFrame jFrame = new JFrame();
       jFrame.setSize(550, 400);
       jFrame.setTitle("Addition Calculator");
       jFrame.setLayout(new GridLayout(3, 3, 10, 10));

       JLabel jLabel = new JLabel("Enter a number");
       jFrame.add(jLabel);
       JTextField jTextField = new JTextField();
       jFrame.add(jTextField);

       JLabel jLabel2 = new JLabel("Enter a number");
       jFrame.add(jLabel2);
       JTextField jTextField2 = new JTextField();
```

```
        jFrame.add(jTextField2);

        JButton jButton = new JButton("Add");
        jFrame.add(jButton);

        JLabel result = new JLabel();

        jButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                try {
                    double num1 = Double.parseDouble(jTextField.getText());
                    double num2 = Double.parseDouble(jTextField2.getText());
                    result.setText("Result is " + (num1 + num2));
                } catch (Exception e) {
                 result.setText("Invalid number");
                 throw new RuntimeException("Invalid Input number");
                }
            }
        });

        jFrame.add(result);

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

Q.Write a program to connect a Java application to a database and perform insert, update, select, and delete operations.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class DBTest {
   public static void main(String[] args) {
       String jdbcURL = "jdbc:mysql://localhost:3306/user";  // Replace with your DB URL
       String username = "root";        // Replace with your DB username
       String password = "";    // Replace with your DB password

       try {
           // 1. Load JDBC Driver
           Class.forName("com.mysql.cj.jdbc.Driver");

           // 2. Establish Connection
           Connection connection = DriverManager.getConnection(jdbcURL, username, password);
           System.out.println("Connected to the database.");

           // 3. Create Statement
           Statement statement = connection.createStatement();

           // // 4. Execute Query
           // String sql = "SELECT id, name FROM test_table";
           // ResultSet resultSet = statement.executeQuery(sql);
           // 4. INSERT Query
           String insertSQL = "INSERT INTO user_login (name, password) VALUES ('Utsav', 'utsav123')";
           int rowsInserted = statement.executeUpdate(insertSQL);
           System.out.println("Rows inserted: " + rowsInserted);

           // // 5. UPDATE Query
           // String updateSQL = "UPDATE test_table SET name = 'Jane Doe' WHERE id = 1";
           // int rowsUpdated = statement.executeUpdate(updateSQL);
           // System.out.println("Rows updated: " + rowsUpdated);

           // 6. SELECT Query
           String selectSQL = "SELECT user_id, name, password FROM user_login";
           ResultSet resultSet = statement.executeQuery(selectSQL);
           // 5. Process Results
           while (resultSet.next()) {
               int id = resultSet.getInt("user_id");
               String name = resultSet.getString("name");
               String userPassword = resultSet.getString("password");
               System.out.println("ID: " + id + ", Name: " + name + ", password
: " + userPassword);
           }

           // 6. Close resources
           resultSet.close();
           statement.close();
           connection.close();
           System.out.println("Connection closed.");

       } catch (Exception e) {
           e.printStackTrace();
       }
   }
}
```

Q.Write a program to build a simple login application using Swing and connect it to a database.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
import java.awt.*;

public class LoginApp {
    public static void main(String[] args) {
       JFrame frame = new JFrame("Login");
       JLabel nameLabel = new JLabel("Username:");
       JLabel passLabel = new JLabel("Password:");
       JTextField nameField = new JTextField(15);
       JPasswordField passField = new JPasswordField(15);
       JButton loginButton = new JButton("Login");

       frame.setLayout(new GridLayout(3, 2, 10, 10));
       frame.add(nameLabel);
       frame.add(nameField);
       frame.add(passLabel);
       frame.add(passField);
       frame.add(new JLabel());
       frame.add(loginButton);

       frame.setSize(500, 400);
       frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       frame.setVisible(true);

       loginButton.addActionListener(e -> {
           String name = nameField.getText();
           String password = new String(passField.getPassword());

           try {
               // Load JDBC driver
               Class.forName("com.mysql.cj.jdbc.Driver");

               // DB connection
               Connection conn = DriverManager.getConnection(
                   "jdbc:mysql://localhost:3306/user", "root", "");

               String sql = "SELECT * FROM user_login WHERE name=? AND
password=?";
               PreparedStatement stmt = conn.prepareStatement(sql);
               stmt.setString(1, name);
               stmt.setString(2, password);
               ResultSet rs = stmt.executeQuery();
               if (rs.next()) {
                   JOptionPane.showMessageDialog(frame, "Login Successful!");
               } else {
                   JOptionPane.showMessageDialog(frame, "Invalid username or 
password");
               }

               rs.close();
               stmt.close();
               conn.close();
           } catch (Exception ex) {
               ex.printStackTrace();
               JOptionPane.showMessageDialog(frame, "Error: " + 
ex.getMessage());
           }
       });
   }
}
```

Q. Write a program to demonstrate the Border Layout in swing. (do it yourself)

Q. Write a program to demonstrate the Group Layout in swing. (do it yourself)

Q. Write a Java program to demonstrate the use of JToolBar in a Swing application. Your program should meet the following requirements:

- Create a JFrame titled "Simple JToolBar Example".
- Add a JToolBar at the top (NORTH) of the frame using BorderLayout.
- Add three buttons to the toolbar labeled New, Open, and Save.
- Set tooltips for each button to describe their action.
- When a button is clicked, display a message dialog showing which button was clicked (e.g., “New clicked”).

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JToolBar;
import java.awt.*;

public class JToolBarExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Simple JToolBar Example");
            frame.setSize(400, 300);

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            // Create toolbar
            JToolBar toolBar = new JToolBar();

            // Create buttons
            JButton newButton = new JButton("New");
            newButton.setToolTipText("Create a new file");

            JButton openButton = new JButton("Open");
            openButton.setToolTipText("Open a file");

            JButton saveButton = new JButton("Save");
            saveButton.setToolTipText("Save the file");

            // Add action listeners
            newButton.addActionListener(e ->
JOptionPane.showMessageDialog(frame, "New clicked"));
            openButton.addActionListener(e ->
JOptionPane.showMessageDialog(frame, "Open clicked"));
```

```
            saveButton.addActionListener(e ->
JOptionPane.showMessageDialog(frame, "Save clicked"));

            // Add buttons to toolbar
            toolBar.add(newButton);
            toolBar.add(openButton);
            toolBar.add(saveButton);

            // Add toolbar and text area to frame
            frame.add(toolBar, BorderLayout.NORTH);
            // Show the frame
            frame.setVisible(true);
    }
}
```

Q. Write a Java program to demonstrate the use of Mnemonics and Accelerators in a Swing Menu.

```
import java.awt.event.InputEvent;
import java.awt.event.KeyEvent;

import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.KeyStroke;

public class MNewmonicsAcceleratorEcmpale {
    public static void main(String[] args) {
         JFrame frame = new JFrame("Mnemonic & Accelerator
Example");

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setSize(400, 300);
            frame.setLocationRelativeTo(null);

            // Create a menu bar
            JMenuBar menuBar = new JMenuBar();
            // Create "File" menu with mnemonic Alt + F
            JMenu jmFile = new JMenu("File");
            jmFile.setMnemonic(KeyEvent.VK_F);  // Alt + F to
open File menu

            // "Open" menu item with mnemonic 'O' and
accelerator Ctrl + O
            JMenuItem jmiOpen = new JMenuItem("Open",
KeyEvent.VK_O);

jmiOpen.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O,
InputEvent.CTRL_DOWN_MASK));
            jmiOpen.addActionListener(e ->
JOptionPane.showMessageDialog(frame, "Open selected"));

            // "Close" menu item with mnemonic 'C' and
accelerator Ctrl + C
            JMenuItem jmiClose = new JMenuItem("Close",
KeyEvent.VK_C);

jmiClose.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_C,
InputEvent.CTRL_DOWN_MASK));
            jmiClose.addActionListener(e ->
JOptionPane.showMessageDialog(frame, "Close selected"));

            // "Save" menu item with mnemonic 'S' and
accelerator Ctrl + S
            JMenuItem jmiSave = new JMenuItem("Save",
KeyEvent.VK_S);

jmiSave.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_S,
InputEvent.CTRL_DOWN_MASK));
            jmiSave.addActionListener(e ->
JOptionPane.showMessageDialog(frame, "Save selected"));

            // "Exit" menu item with mnemonic 'E' and
accelerator Ctrl + E
            JMenuItem jmiExit = new JMenuItem("Exit",
KeyEvent.VK_E);
```

```
jmiExit.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_E,
InputEvent.CTRL_DOWN_MASK));
            jmiExit.addActionListener(e -> System.exit(0));

            // Add menu items to menu
            jmFile.add(jmiOpen);
            jmFile.add(jmiClose);
            jmFile.add(jmiSave);
            jmFile.addSeparator();
            jmFile.add(jmiExit);

            // Add menu to menu bar
            menuBar.add(jmFile);

            // Set menu bar in frame
            frame.setJMenuBar(menuBar);

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

Q. Write a Java program to demonstrate the use of JTree in a Swing application. Your program should meet the following requirements:

- Create a JFrame titled "JTree Example".
- Construct a tree structure using DefaultMutableTreeNode

```
import javax.swing.JFrame;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;

public class TreeExample {
     public static void main(String[] args) {
        JFrame frame = new JFrame("JTree Example");

        // Creating tree nodes
        DefaultMutableTreeNode root = new
DefaultMutableTreeNode("Root");
        DefaultMutableTreeNode folder1 = new
DefaultMutableTreeNode("Folder1");
        DefaultMutableTreeNode folder2 = new
DefaultMutableTreeNode("Folder2");
        DefaultMutableTreeNode folder3 = new
DefaultMutableTreeNode("Folder3");

        root.add(folder1);
        root.add(folder2);
        folder1.add(folder3);
        folder3.add(new DefaultMutableTreeNode("File3.txt"));
        folder1.add(new DefaultMutableTreeNode("File1.txt"));
        folder2.add(new DefaultMutableTreeNode("File2.txt"));

        JTree tree = new JTree(root);

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

Q. Write a Java program to demonstrate the use of JInternalFrame in a Swing application.

```
import javax.swing.*;

public class JInternalFrameExample {
    public static void main(String[] args)
    {
        JFrame mainFrame = new JFrame("JInternalFrame Example");
        mainFrame.setSize(600, 400);

mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JInternalFrame internalFrame = new
JInternalFrame("Internal Window", true, true, true, true);
```

```
        internalFrame.setSize(300, 200);
        internalFrame.setVisible(true);
        internalFrame.add(new JLabel("This is an internal
frame"));

        mainFrame.add(internalFrame);

        mainFrame.setVisible(true);
    }

}
```