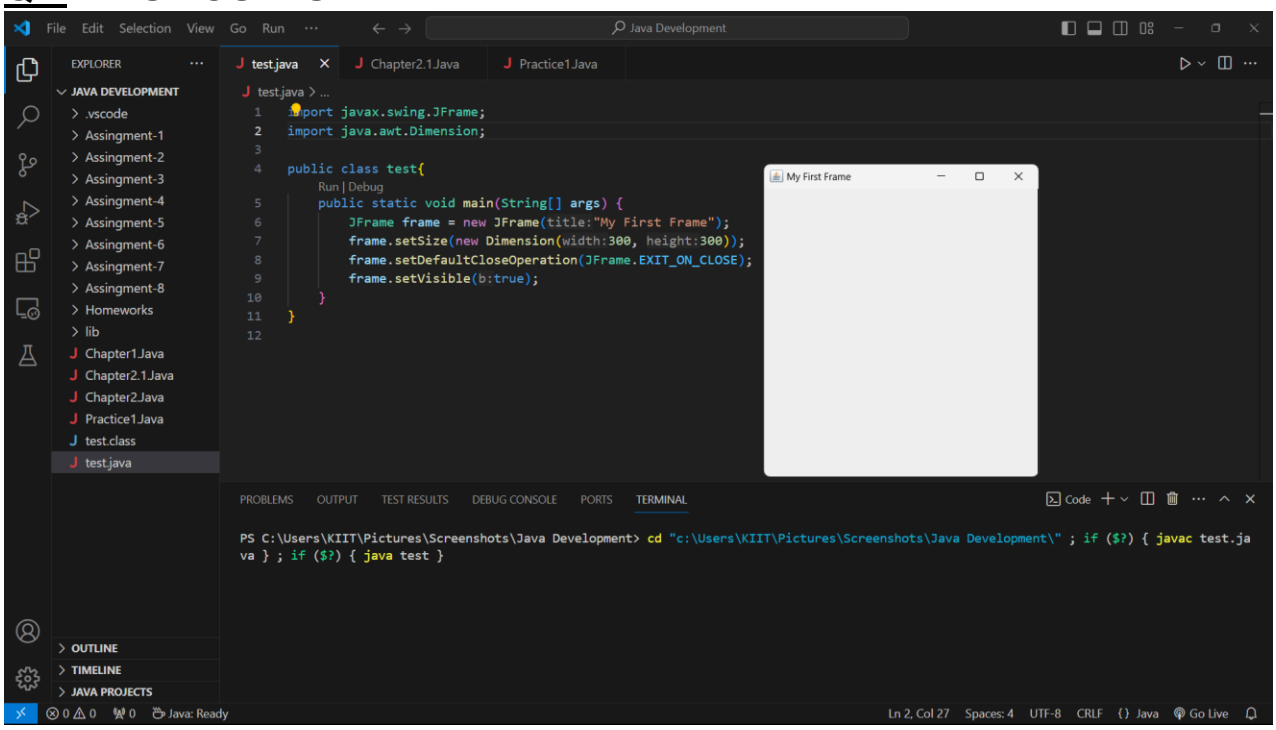
ASSINGMENT-4

SUBMITTED BY: - **KANISHK**

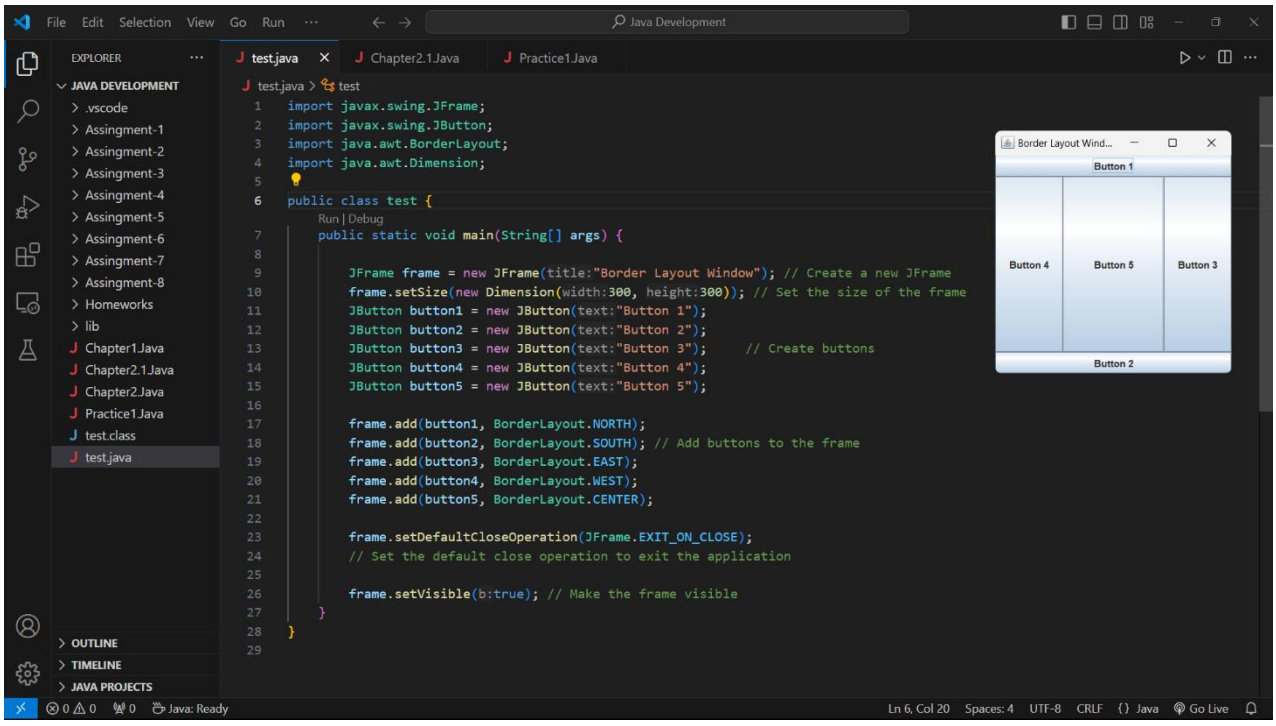
2205130

CSE-37

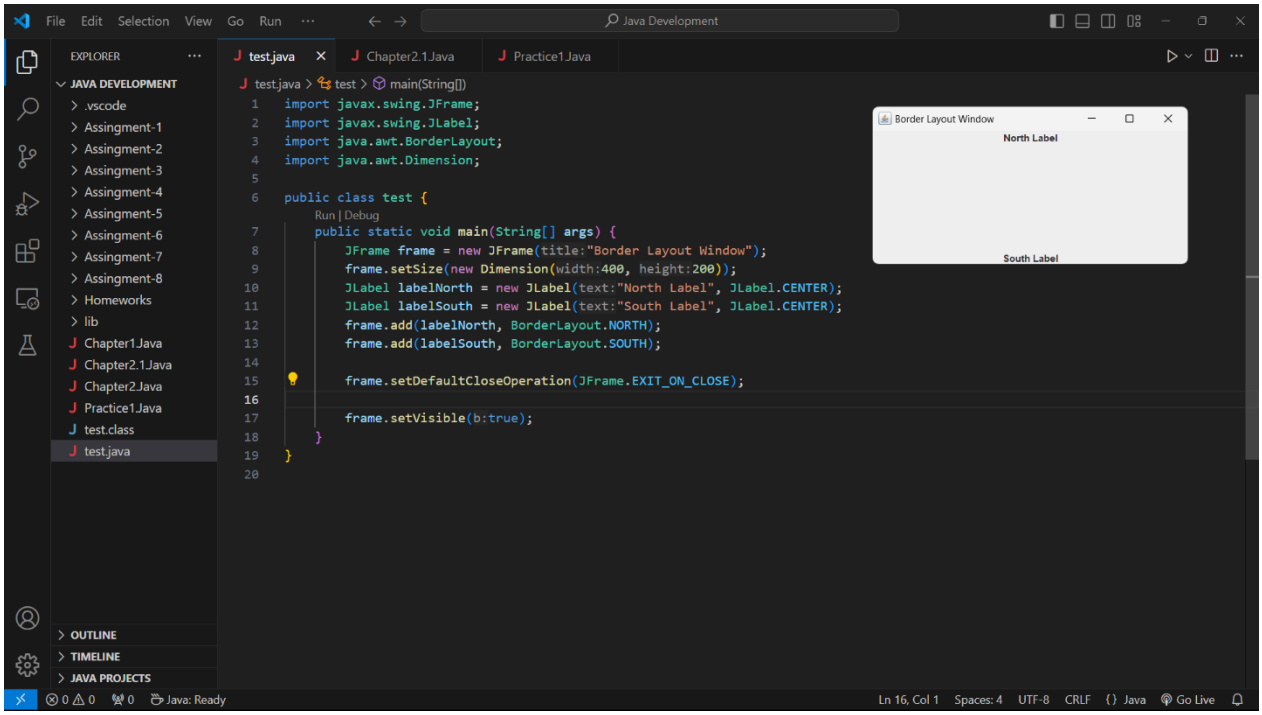
Q1- INPUT/OUTPUT



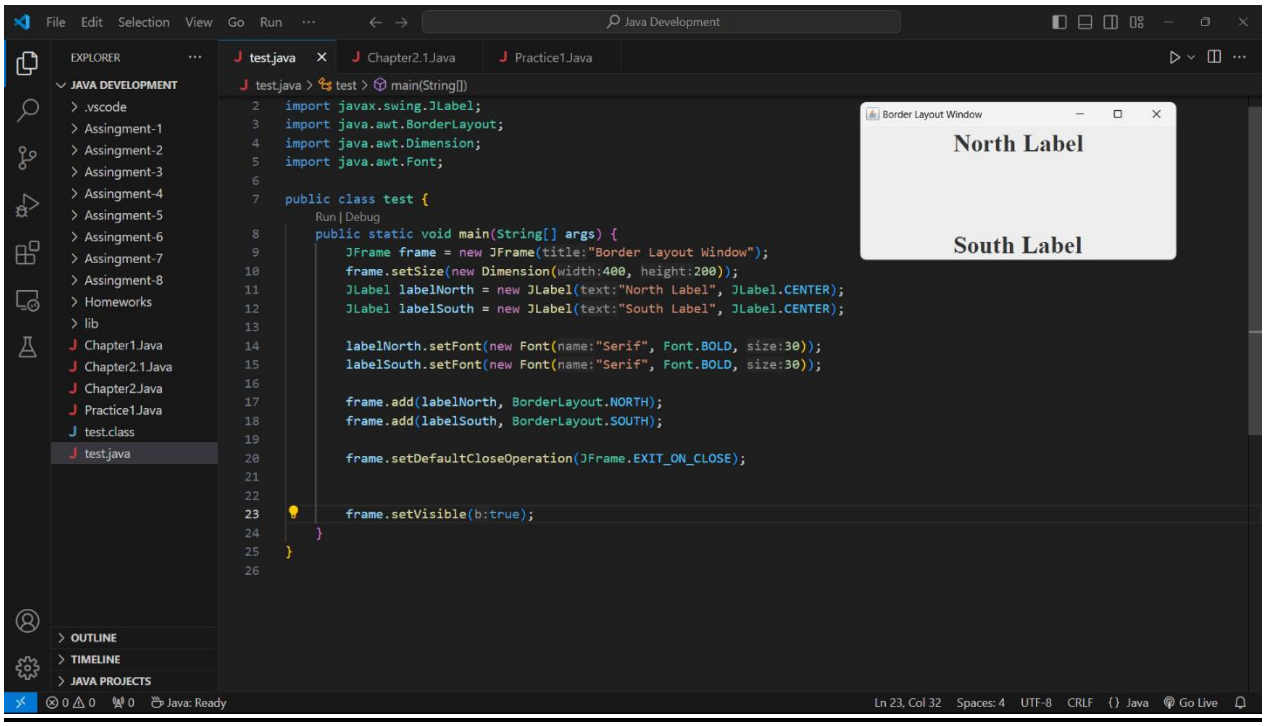
Q2: - INPUT/OUTPUT



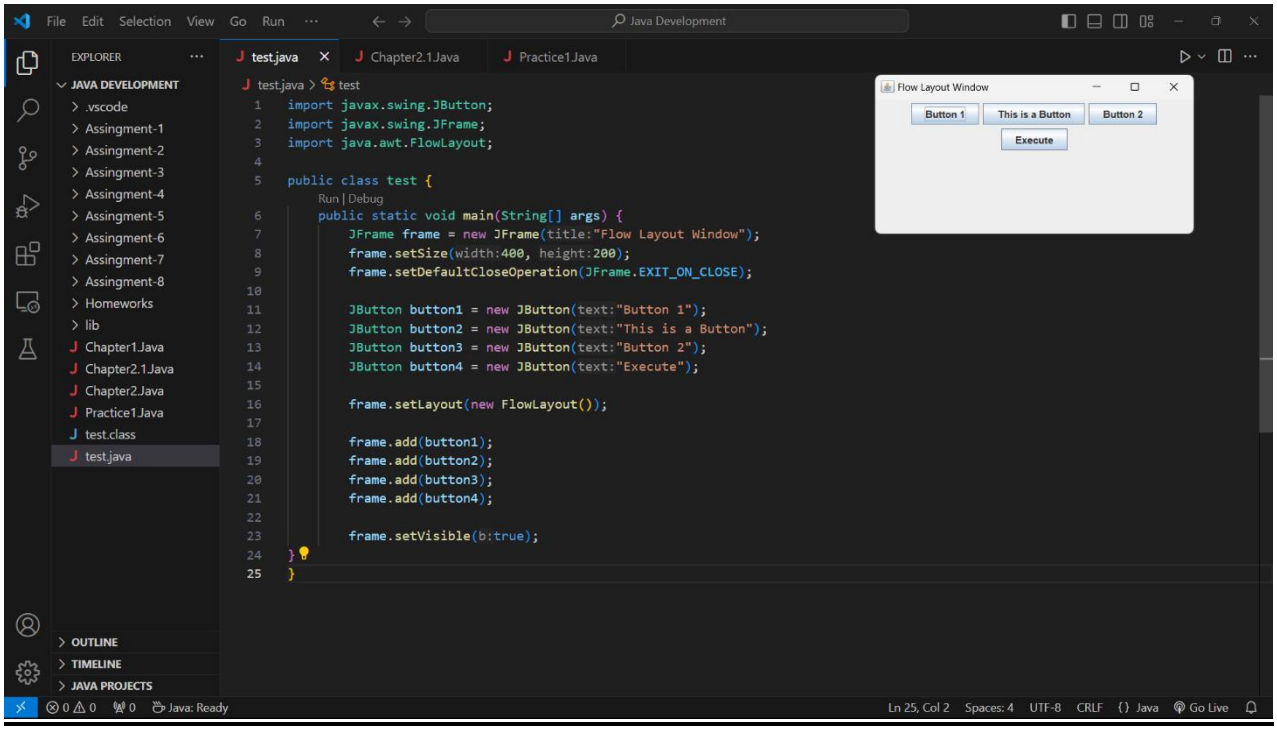
Q3: - INPUT/OUTPUT



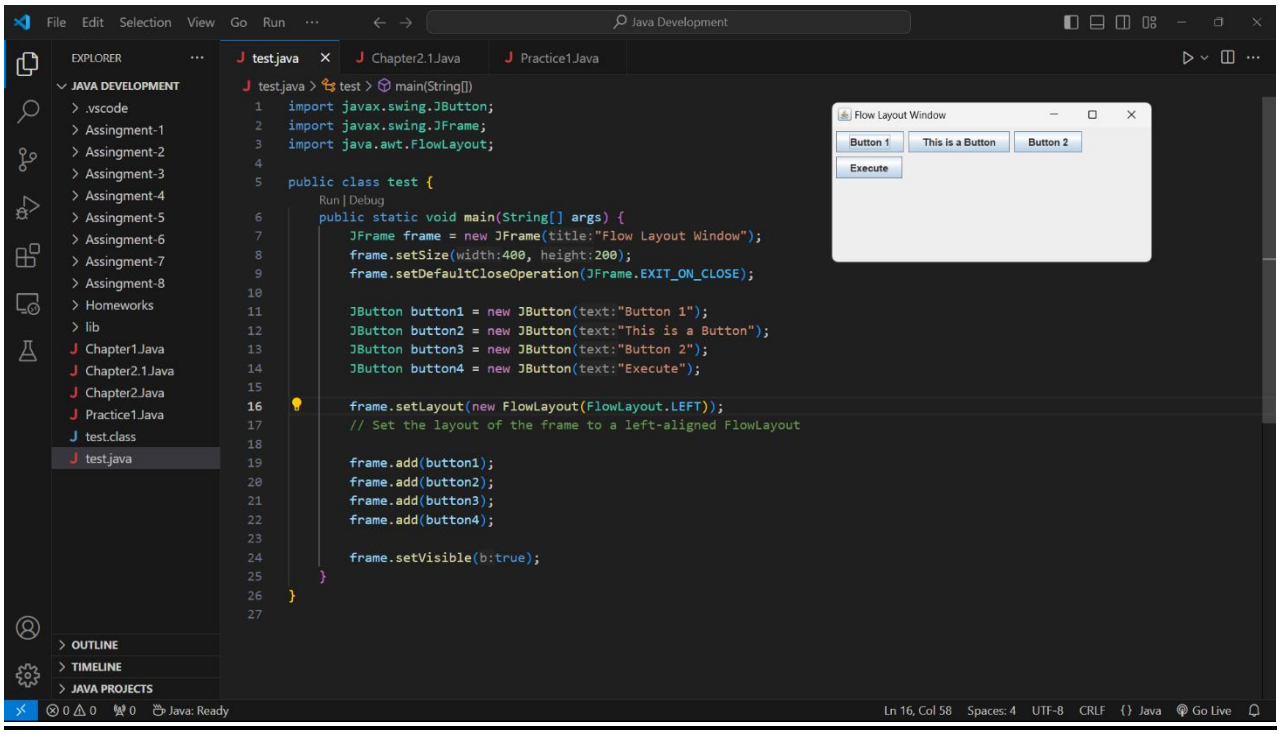
Q4: - INPUT/OUTPUT



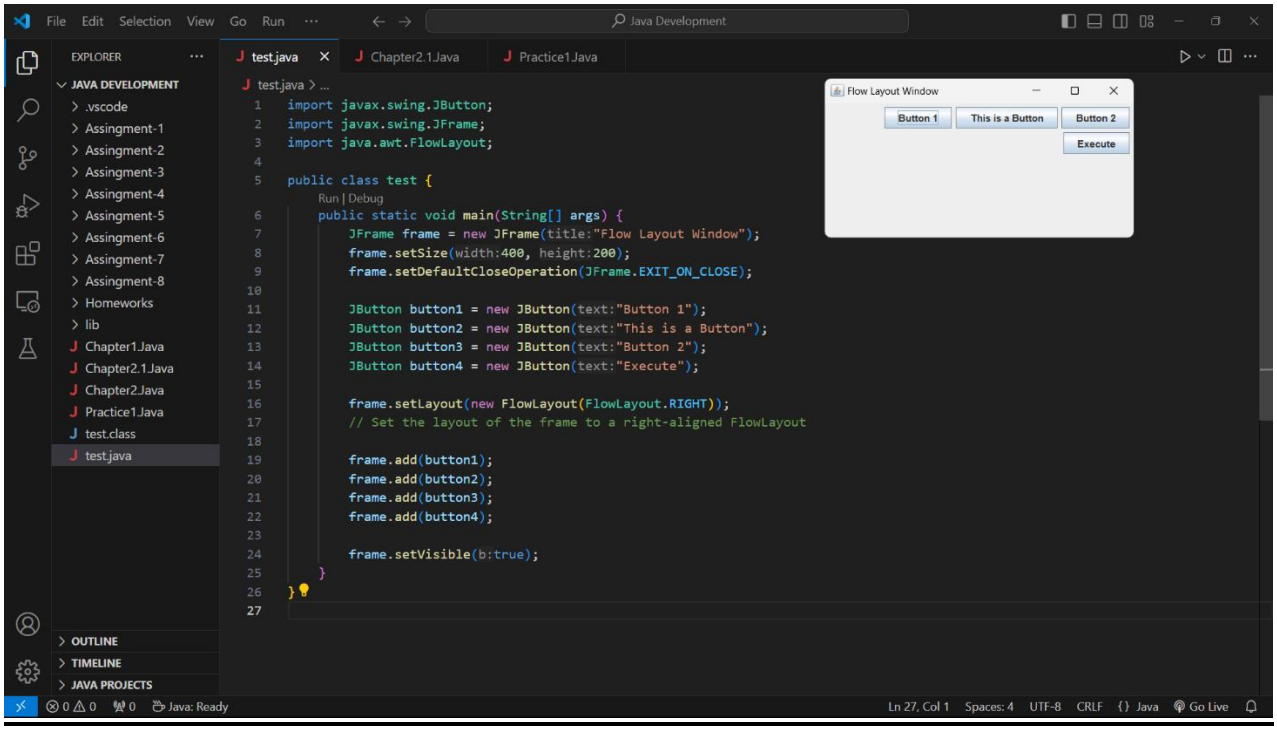
Q5: - INPUT/OUTPUT



Q6: - INPUT/OUTPUT

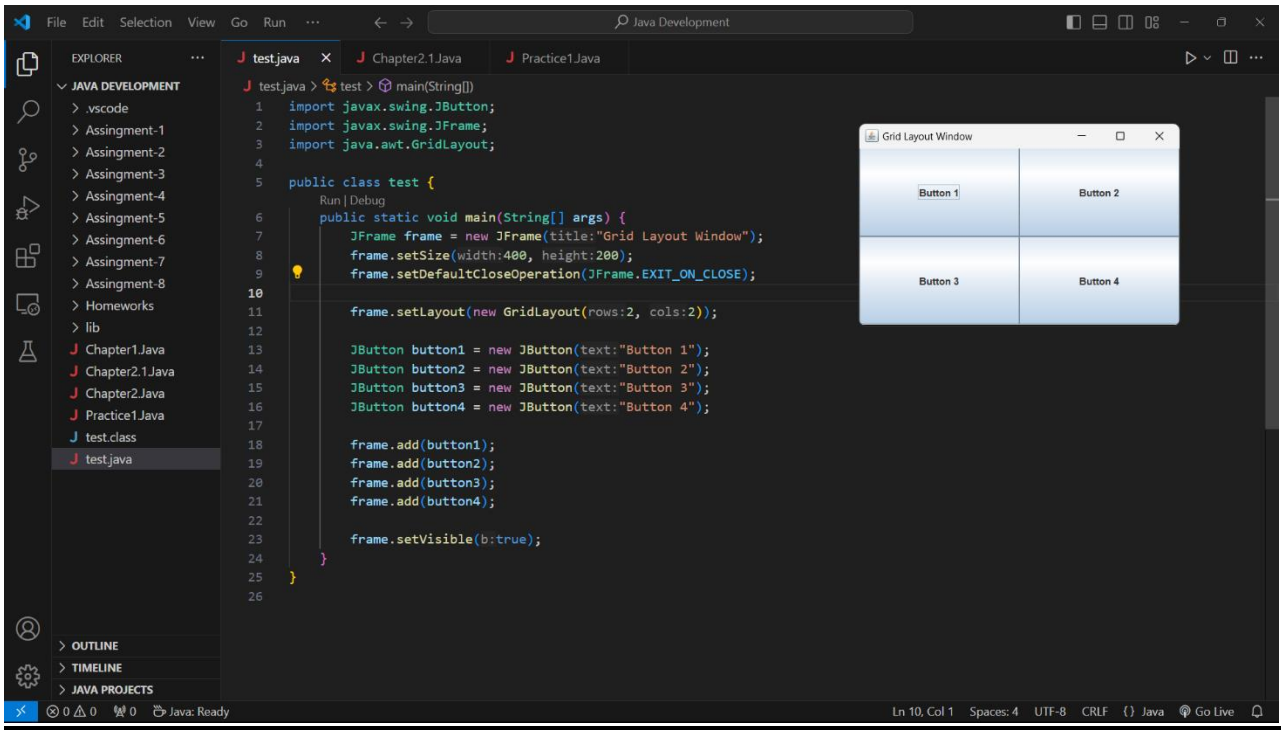


Q7: - INPUT/OUTPUT




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

public class test {
    Run | Debug
    public static void main(String[] args) {
        JFrame frame = new JFrame(title:"Flow Layout Window");
        frame.setSize(width:400, height:200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JButton button1 = new JButton(text:"Button 1");
        JButton button2 = new JButton(text:"This is a Button");
        JButton button3 = new JButton(text:"Button 2");
        JButton button4 = new JButton(text:"Execute");

        frame.setLayout(new FlowLayout(FlowLayout.RIGHT));
        // Set the layout of the frame to a right-aligned FlowLayout

        frame.add(button1);
        frame.add(button2);
        frame.add(button3);
        frame.add(button4);

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

Q8: - INPUT/OUTPUT



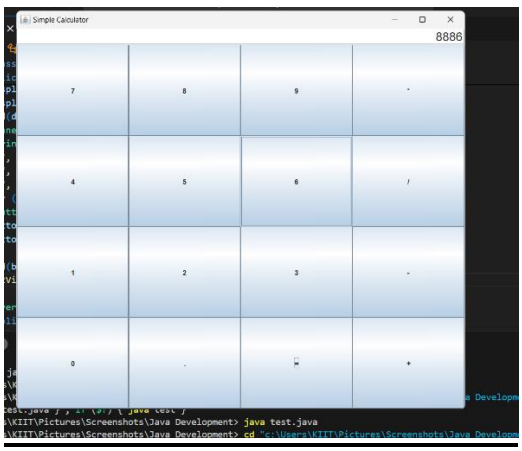

```java
import javax.swing.JButton;
import javax.swing.JFrame;
import java.awt.GridLayout;

public class test {
    Run | Debug
    public static void main(String[] args) {
        JFrame frame = new JFrame(title:"Grid Layout Window");
        frame.setSize(width:400, height:200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setLayout(new GridLayout(rows:2, cols:2));

        JButton button1 = new JButton(text:"Button 1");
        JButton button2 = new JButton(text:"Button 2");
        JButton button3 = new JButton(text:"Button 3");
        JButton button4 = new JButton(text:"Button 4");

        frame.add(button1);
        frame.add(button2);
        frame.add(button3);
        frame.add(button4);

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

Q9: - INPUT: -

```java
package GUI;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
class CalculatorGUI extends JFrame implements ActionListener {
private JTextField displayField;
private double currentValue;
private char currentOperator;
public CalculatorGUI() {
super(title:"Simple Calculator");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(width:300, height:300);
displayField = new JTextField();
displayField.setFont(new Font(name:"Arial", Font.PLAIN, size:20));
displayField.setHorizontalAlignment(JTextField.RIGHT);
add(displayField, BorderLayout.NORTH);
JPanel buttonPanel = new JPanel(new GridLayout(rows:4, cols:4));
String[] buttonLabels = {"7", "8", "9", "*",
"4", "5", "6", "/",
"1", "2", "3", "-",
"0", ".", "=", "+"};
for (String label : buttonLabels) {
JButton button = new JButton(label);
button.addActionListener(this);
buttonPanel.add(button);
}
add(buttonPanel, BorderLayout.CENTER);
setVisible(b:true);
}
@Override
```

```java
class CalculatorGUI extends JFrame implements ActionListener {
public CalculatorGUI() {
"1", "2", "3", "-",
"0", ".", "=", "+"};
for (String label : buttonLabels) {
JButton button = new JButton(label);
button.addActionListener(this);
buttonPanel.add(button);
}
add(buttonPanel, BorderLayout.CENTER);
setVisible(b:true);
}
@Override
public void actionPerformed(ActionEvent e) {
String command = e.getActionCommand();
if (Character.isDigit(command.charAt(index:0))) {
displayField.setText(displayField.getText() + command);
} else if (command.equals(anObject:".")) {
} else if (command.equals(anObject:"=")) {
} else {
}
}
Run | Debug
public static void main(String[] args) {
SwingUtilities.invokeLater(()-> new CalculatorGUI());
}}
```

OUTPUT: -

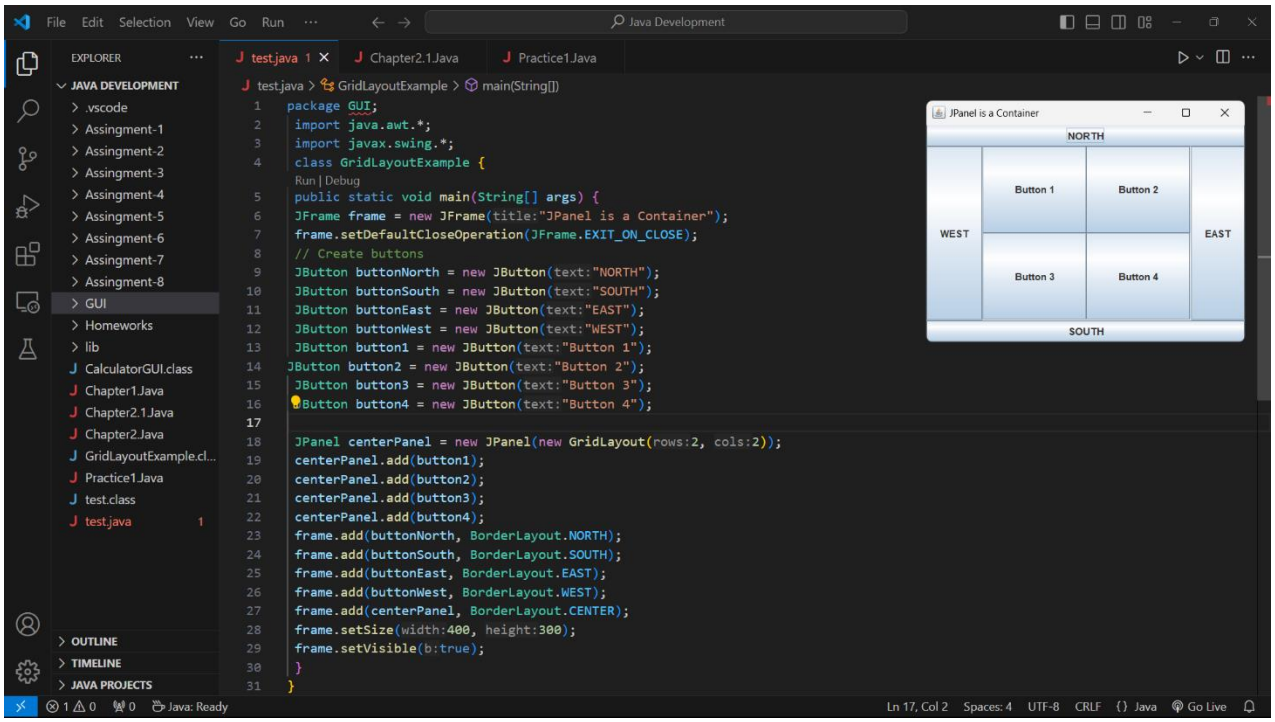


Q10: - The given program creates a JFrame with a JPanel inside it. The JPanel uses the default layout, which is Flow Layout. Flow Layout arranges components in a horizontal row, with each component's width set to its preferred width. If the horizontal space in the container is too small to fit all the components in one row, Flow Layout will wrap the extra components to the next row.

In this case, the JPanel contains four JButton components. Each JButton has a preferred width that is determined by the text and the font size of the button. The JPanel's width is set to 200, which is not enough to fit all the buttons in one row. Therefore, the Flow Layout will wrap the extra buttons to the next row.

TheoutputoftheprogramwillbeawindowwithaJPanelthatcontainsfourJButton components. The buttons will be arranged in two rows, with two buttons in the first row and two buttons in these condrow. The window will be resizable, and the user can close it.

Q11: - INPUT/OUTPUT



Q12: - In the given JPanel example, the JFrame contains a JPanel named "buttonPanel". The JPanel uses the default layout, which is FlowLayout. FlowLayout arranges components in a horizontal row, with each component's width set to its preferred width.

The "buttonPanel" contains three JButton components. Each JButton has a preferred width that is determined by the text and the font size of the button. The JPanel's width is

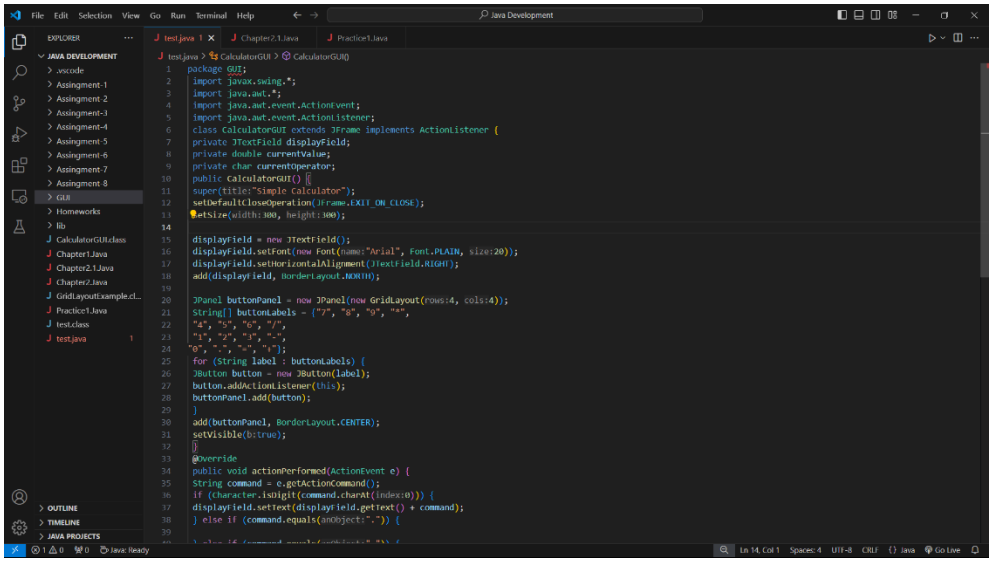
determined by the width of the JFrame, which is 300 pixels.

The "buttonPanel" is added to the JFrame using the method add ("Center", buttonPanel). This method adds the "buttonPanel" to the JFrame's content pane with the constraint "Center". The constraint "Center" tells the layout manager to place the "buttonPanel" in the center of the JFrame.

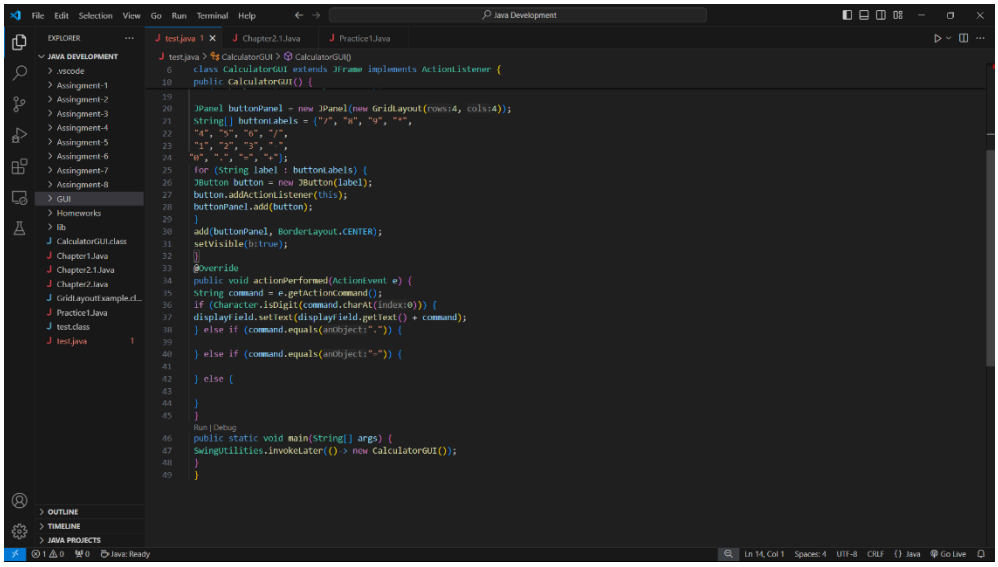
TheJFrame'slayoutmanagerisaBorderLayoutbydefault.BorderLayoutplaces Components in five regions: North, South, East, West, and Center. The "buttonPanel" is Placed in the Center region of the JFrame.

TheoutputoftheprogramwillbeawindowwithaJPanelthatcontainsthreeJButton components. The buttons will be arranged in a single row, with Each button's width set to its preferred width. The window will be resizable, and the user can close it.

Q13: - INPUT: -

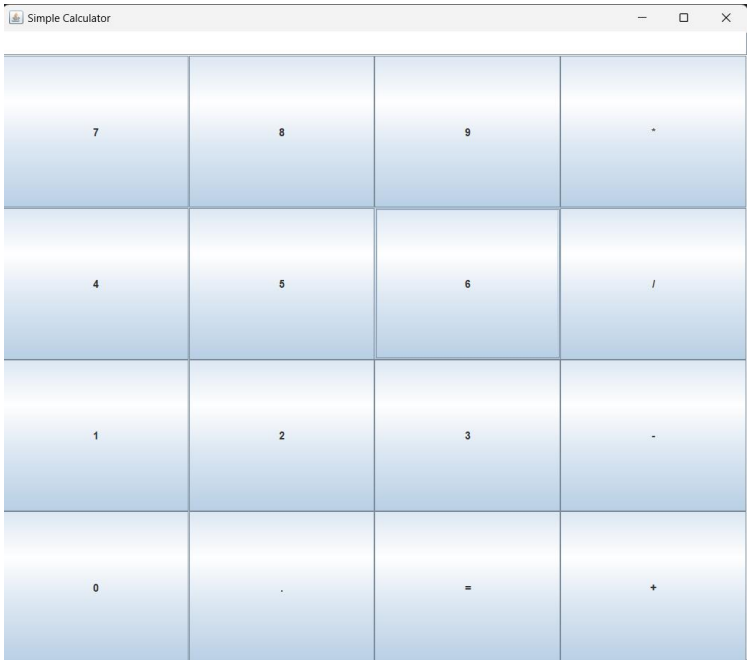


```
package GUI;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
class CalculatorGUI extends JFrame implements ActionListener {
private JTextField displayField;
private double currentValue;
private char currentOperator;
public CalculatorGUI() {
super(title:"Simple Calculator");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(width:300, height:300);

displayField = new JTextField();
displayField.setFont(new Font(name:"Arial", Font.PLAIN, size:20));
displayField.setHorizontalAlignment(JTextField.RIGHT);
add(displayField, BorderLayout.NORTH);

JPanel buttonPanel = new JPanel(new GridLayout(rows:4, cols:4));
String[] buttonLabels = {"7", "8", "9", "*",
"4", "5", "6", "/",
"1", "2", "3", "-",
"0", ".", "=", "+"};
for (String label : buttonLabels) {
JButton button = new JButton(label);
button.addActionListener(this);
buttonPanel.add(button);
}
add(buttonPanel, BorderLayout.CENTER);
setVisible(b:true);
}
@Override
public void actionPerformed(ActionEvent e) {
String command = e.getActionCommand();
if (Character.isDigit(command.charAt(index:0))) {
displayField.setText(displayField.getText() + command);
} else if (command.equals(anObject:".")) {
```

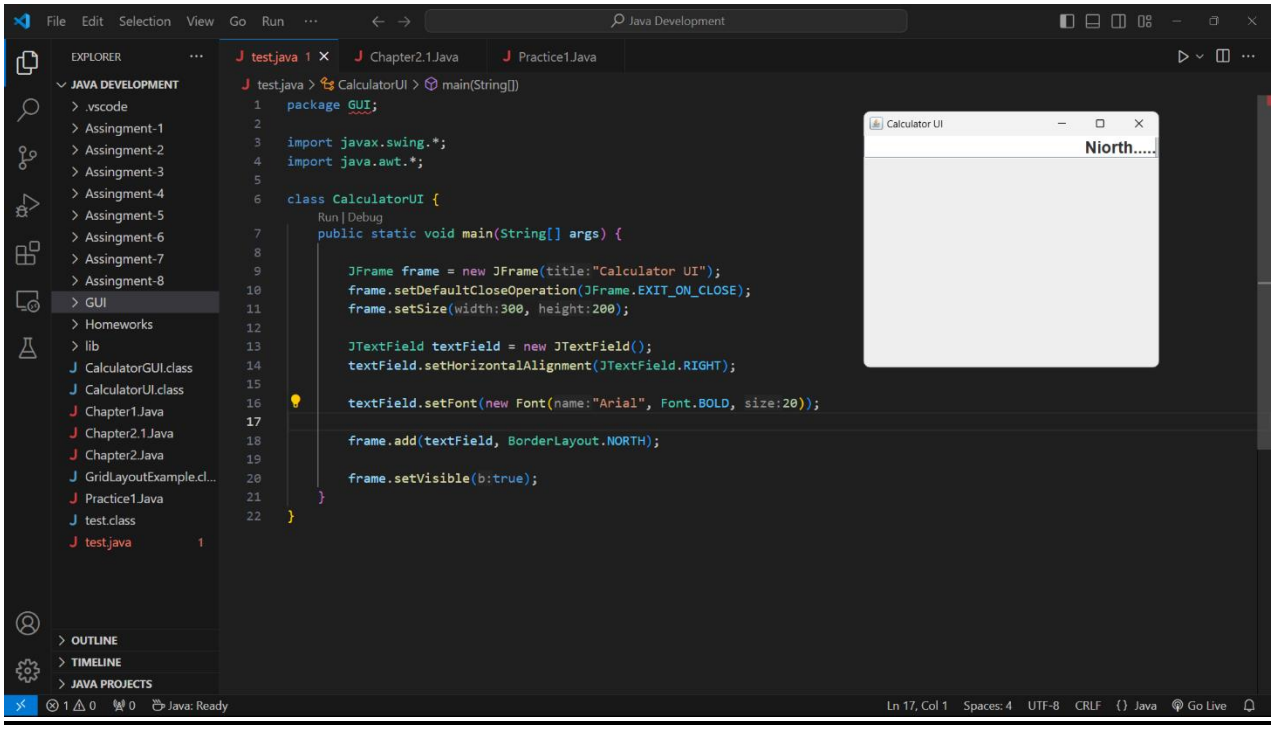


```
class CalculatorGUI extends JFrame implements ActionListener {
public CalculatorGUI() {

JPanel buttonPanel = new JPanel(new GridLayout(rows:4, cols:4));
String[] buttonLabels = {"7", "8", "9", "*",
"4", "5", "6", "/",
"1", "2", "3", "-",
"0", ".", "=", "+"};
for (String label : buttonLabels) {
JButton button = new JButton(label);
button.addActionListener(this);
buttonPanel.add(button);
}
add(buttonPanel, BorderLayout.CENTER);
setVisible(b:true);
}
@Override
public void actionPerformed(ActionEvent e) {
String command = e.getActionCommand();
if (Character.isDigit(command.charAt(index:0))) {
displayField.setText(displayField.getText() + command);
} else if (command.equals(anObject:".")) {

} else if (command.equals(anObject:"-")) {

} else {

}
}
Run | Debug
public static void main(String[] args) {
SwingUtilities.invokeLater(() -> new CalculatorGUI());
}
}
```

OUTPUT: -



Q14: - INPUT/OUTPUT



Q15: - The layout of the given GUI can be explained as follows:

1. TheJFrame's layout manager is set to a FlowLayout. FlowLayout arranges components in a horizontal row, with each component's width set to its preferred width.

2. TheJFrame contains a JTextField (num1textField), a JLabel (operatorLabel),

another JTextField (num2textField), a JButton (equalsButton), and a JLabel (answerLabel).

3. TheJTextField (num1textField) is used to input the first number. It has a preferred width of 5 characters.

4. TheJLabel (operatorLabel) displays the operator (+) between the two numbers.

5. TheJTextField (num2textField) is used to input the second number. It also has a preferred width of 5 characters.

6. TheJButton (equalsButton) is used to calculate the sum of the two numbers.

7. TheJLabel (answerLabel) displays the result of the calculation.

8. TheJFrame's size is set to 300x300 pixels using the setSize() method.

9. TheJFrame's title is set to "Calculator" using the setTitle() method.

10. The JFrame is made resizable and closable using the setDefaultCloseOperation() method with the argument EXIT_ON_CLOSE.

11. The JFrame is centered on the screen using the setLocationRelativeTo() method with the argument null.

12. The JFrame is made visible using the setVisible() method with the argument true.

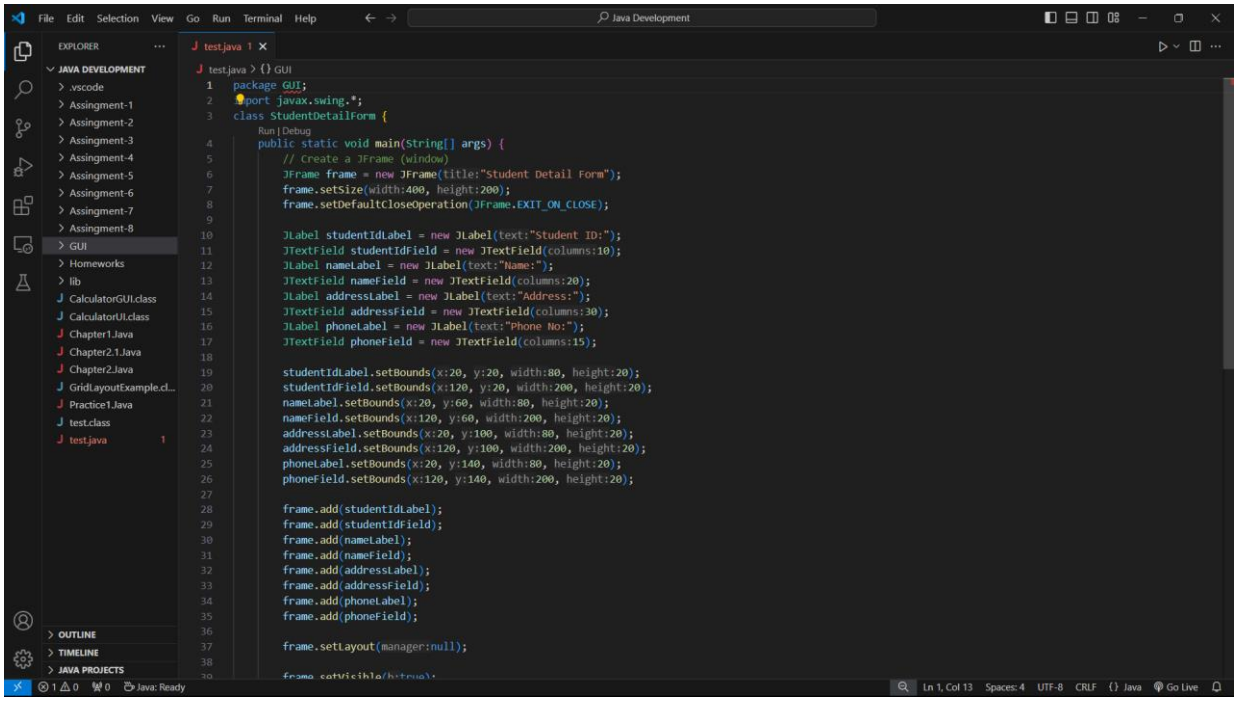
13. The pack() method is called to adjust the size of the JFrame to fit the preferred sizes of its components.

Q16: - The output of the program will be a window with a JTextField for inputting the first number, a JLabel for displaying the operator, another JTextField for inputting the second number, a JButton for calculating the sum, and a JLabel for displaying the result. The window will be resizable and closable.

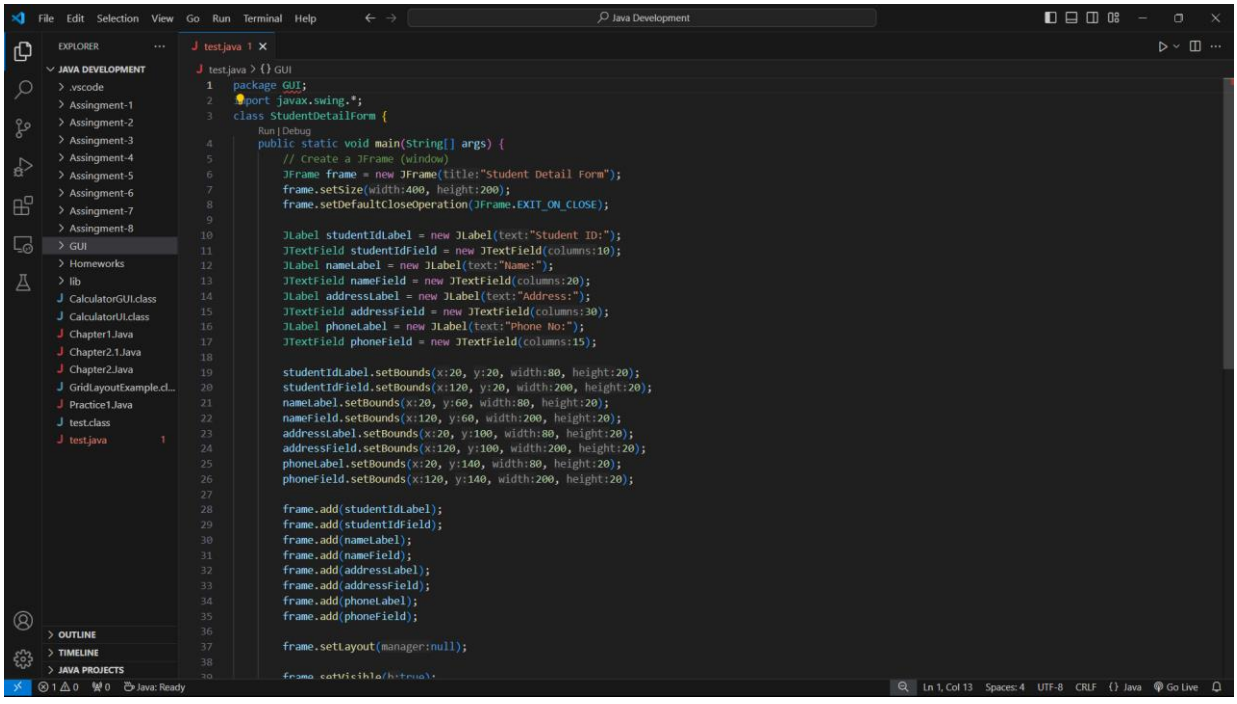
Q17: - The output of the program will be a window with a JLabel displaying the text "This is a JLabel" and an image of a Facebook logo. The JLabel is horizontally centered and uses the font "Batang" with a size of 20. The window will be resizable and closable. If the file "facebook.png" is not in the working directory, the program will throw an exception and the window will not display the image.

Q19: - The output of the program will be a window with a JComboBox displaying a dropdown list of colors: RED, GREEN, BLACK, and BLUE. The user can select a color from the dropdown list. The selected color will be displayed in the JComboBox.

Q20: - INPUT: -

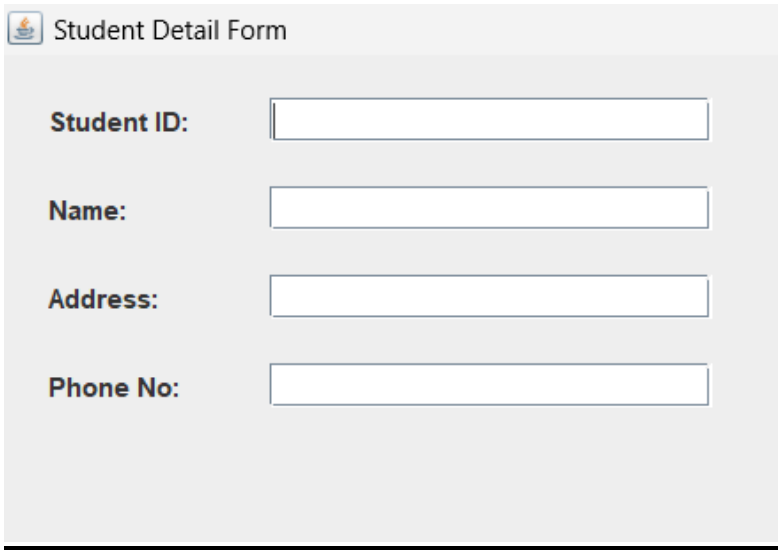


```java
package GUI;
import javax.swing.*;
class StudentDetailForm {
    Run | Debug
    public static void main(String[] args) {
        // Create a JFrame (window)
        JFrame frame = new JFrame(title:"Student Detail Form");
        frame.setSize(width:400, height:200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel studentIdLabel = new JLabel(text:"Student ID:");
        JTextField studentIdField = new JTextField(columns:10);
        JLabel nameLabel = new JLabel(text:"Name:");
        JTextField nameField = new JTextField(columns:20);
        JLabel addressLabel = new JLabel(text:"Address:");
        JTextField addressField = new JTextField(columns:30);
        JLabel phoneLabel = new JLabel(text:"Phone No:");
        JTextField phoneField = new JTextField(columns:15);

        studentIdLabel.setBounds(x:20, y:20, width:80, height:20);
        studentIdField.setBounds(x:120, y:20, width:200, height:20);
        nameLabel.setBounds(x:20, y:60, width:80, height:20);
        nameField.setBounds(x:120, y:60, width:200, height:20);
        addressLabel.setBounds(x:20, y:100, width:80, height:20);
        addressField.setBounds(x:120, y:100, width:200, height:20);
        phoneLabel.setBounds(x:20, y:140, width:80, height:20);
        phoneField.setBounds(x:120, y:140, width:200, height:20);

        frame.add(studentIdLabel);
        frame.add(studentIdField);
        frame.add(nameLabel);
        frame.add(nameField);
        frame.add(addressLabel);
        frame.add(addressField);
        frame.add(phoneLabel);
        frame.add(phoneField);

        frame.setLayout(manager:null);
```

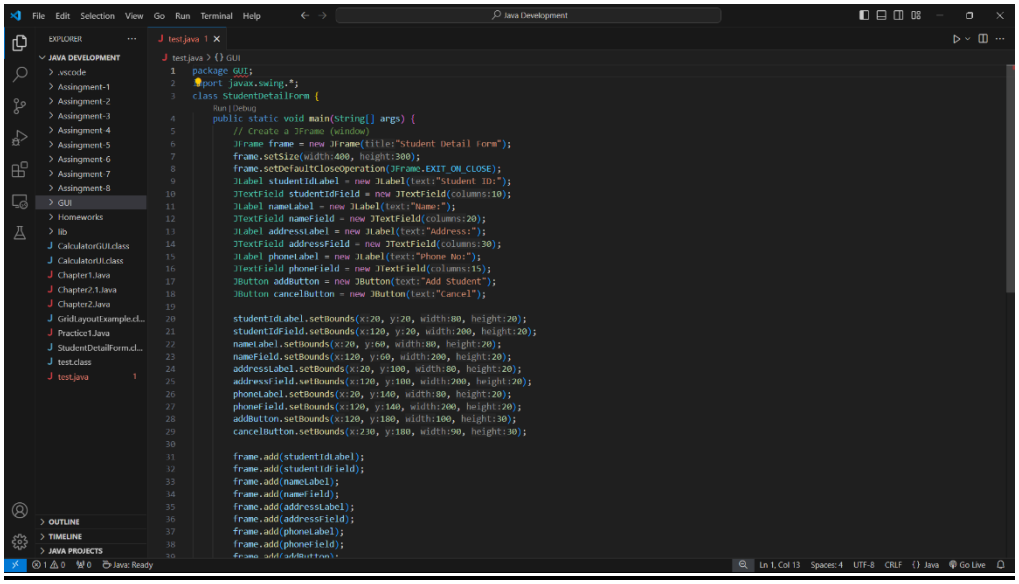



```java
package GUI;
import javax.swing.*;
class StudentDetailForm {
    Run | Debug
    public static void main(String[] args) {
        // Create a JFrame (window)
        JFrame frame = new JFrame(title:"Student Detail Form");
        frame.setSize(width:400, height:200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel studentIdLabel = new JLabel(text:"Student ID:");
        JTextField studentIdField = new JTextField(columns:10);
        JLabel nameLabel = new JLabel(text:"Name:");
        JTextField nameField = new JTextField(columns:20);
        JLabel addressLabel = new JLabel(text:"Address:");
        JTextField addressField = new JTextField(columns:30);
        JLabel phoneLabel = new JLabel(text:"Phone No:");
        JTextField phoneField = new JTextField(columns:15);

        studentIdLabel.setBounds(x:20, y:20, width:80, height:20);
        studentIdField.setBounds(x:120, y:20, width:200, height:20);
        nameLabel.setBounds(x:20, y:60, width:80, height:20);
        nameField.setBounds(x:120, y:60, width:200, height:20);
        addressLabel.setBounds(x:20, y:100, width:80, height:20);
        addressField.setBounds(x:120, y:100, width:200, height:20);
        phoneLabel.setBounds(x:20, y:140, width:80, height:20);
        phoneField.setBounds(x:120, y:140, width:200, height:20);

        frame.add(studentIdLabel);
        frame.add(studentIdField);
        frame.add(nameLabel);
        frame.add(nameField);
        frame.add(addressLabel);
        frame.add(addressField);
        frame.add(phoneLabel);
        frame.add(phoneField);

        frame.setLayout(manager:null);
```

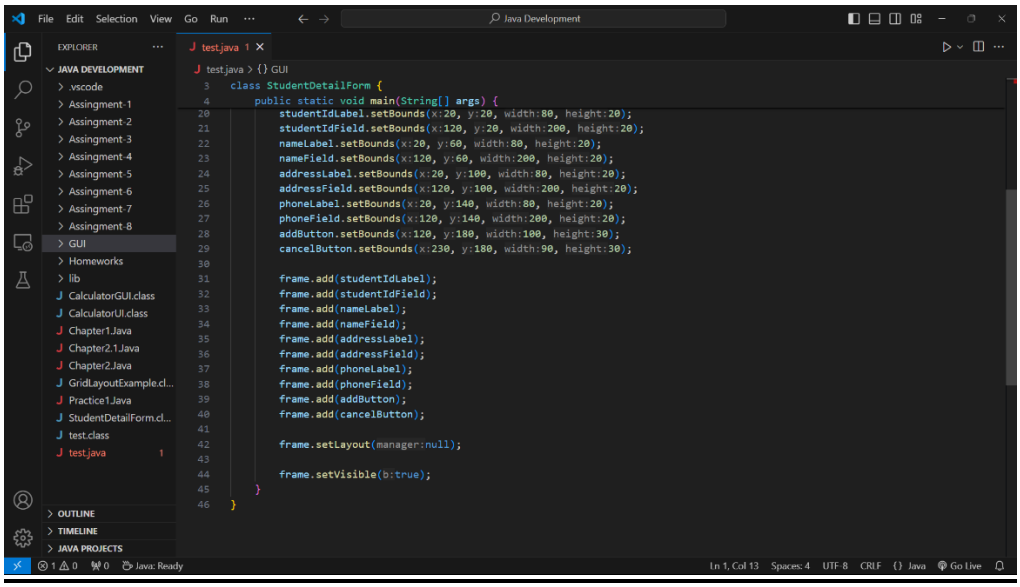

OUTPUT: -



Q21: - INPUT

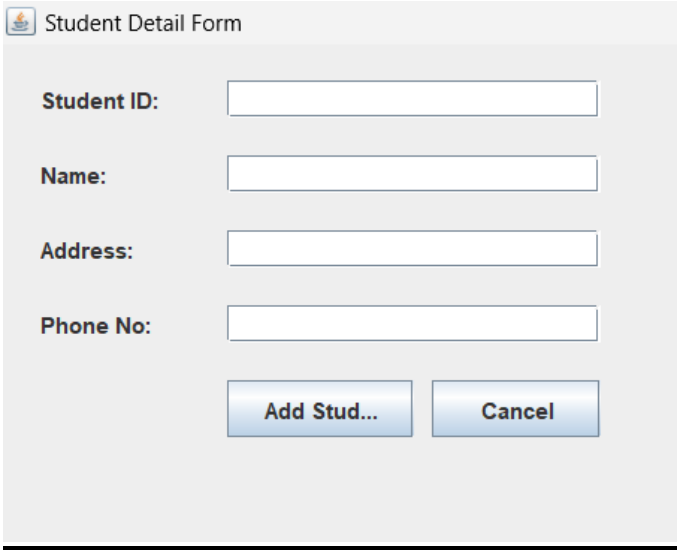


```
package GUI;
import javax.swing.*;
class StudentDetailForm {
    public static void main(String[] args) {
        // Create a JFrame (window)
        JFrame frame = new JFrame(title:"Student Detail Form");
        frame.setSize(width:400, height:300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel studentIdLabel = new JLabel(text:"Student ID:");
        JTextField studentIdField = new JTextField(columns:10);
        JLabel nameLabel = new JLabel(text:"Name:");
        JTextField nameField = new JTextField(columns:20);
        JLabel addressLabel = new JLabel(text:"Address:");
        JTextField addressField = new JTextField(columns:30);
        JLabel phoneLabel = new JLabel(text:"Phone No:");
        JTextField phoneField = new JTextField(columns:15);
        JButton addButton = new JButton(text:"Add Student");
        JButton cancelButton = new JButton(text:"Cancel");

        studentIdLabel.setBounds(x:20, y:20, width:80, height:20);
        studentIdField.setBounds(x:120, y:20, width:200, height:20);
        nameLabel.setBounds(x:20, y:60, width:80, height:20);
        nameField.setBounds(x:120, y:60, width:200, height:20);
        addressLabel.setBounds(x:20, y:100, width:80, height:20);
        addressField.setBounds(x:120, y:100, width:200, height:20);
        phoneLabel.setBounds(x:20, y:140, width:80, height:20);
        phoneField.setBounds(x:120, y:140, width:200, height:20);
        addButton.setBounds(x:120, y:180, width:100, height:30);
        cancelButton.setBounds(x:230, y:180, width:90, height:30);

        frame.add(studentIdLabel);
        frame.add(studentIdField);
        frame.add(nameLabel);
        frame.add(nameField);
        frame.add(addressLabel);
        frame.add(addressField);
        frame.add(phoneLabel);
        frame.add(phoneField);
```



```
class StudentDetailForm {
    public static void main(String[] args) {
        studentIdLabel.setBounds(x:20, y:20, width:80, height:20);
        studentIdField.setBounds(x:120, y:20, width:200, height:20);
        nameLabel.setBounds(x:20, y:60, width:80, height:20);
        nameField.setBounds(x:120, y:60, width:200, height:20);
        addressLabel.setBounds(x:20, y:100, width:80, height:20);
        addressField.setBounds(x:120, y:100, width:200, height:20);
        phoneLabel.setBounds(x:20, y:140, width:80, height:20);
        phoneField.setBounds(x:120, y:140, width:200, height:20);
        addButton.setBounds(x:120, y:180, width:100, height:30);
        cancelButton.setBounds(x:230, y:180, width:90, height:30);

        frame.add(studentIdLabel);
        frame.add(studentIdField);
        frame.add(nameLabel);
        frame.add(nameField);
        frame.add(addressLabel);
        frame.add(addressField);
        frame.add(phoneLabel);
        frame.add(phoneField);
        frame.add(addButton);
        frame.add(cancelButton);

        frame.setLayout(manager:null);

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

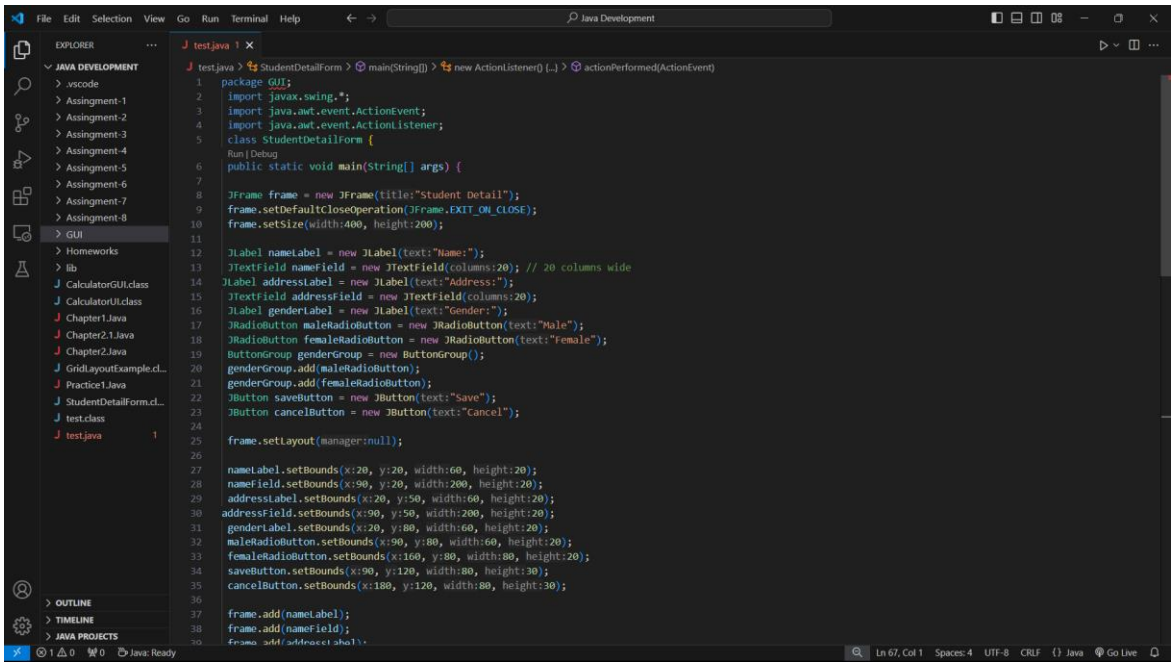
OUTPUT: -

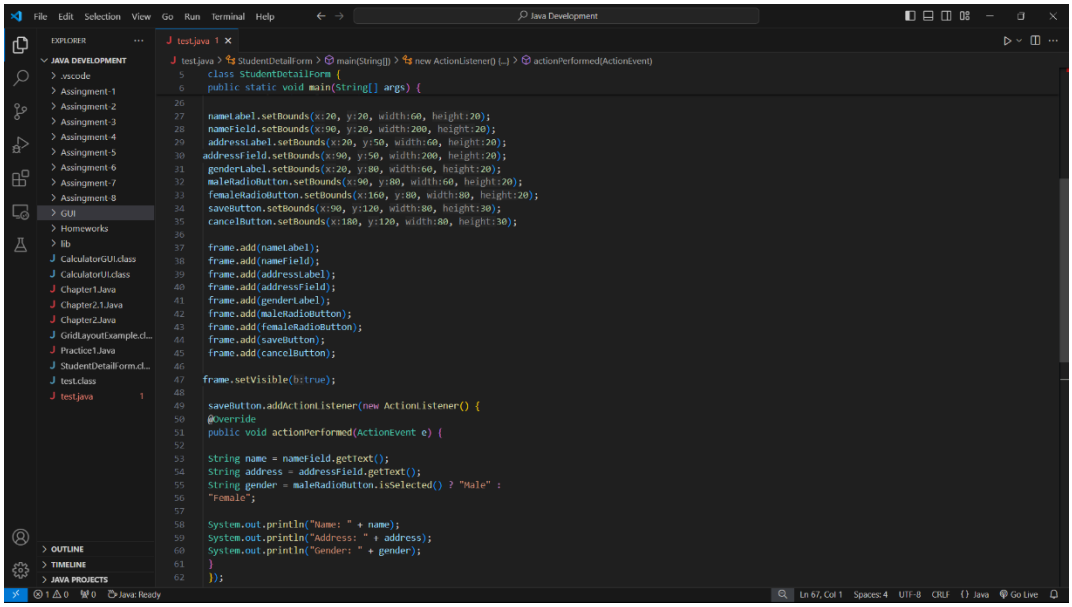


Q22: - A toggle button, represented by the JToggleButton class in Java, is a button that can be either selected or deselected. When a toggle button is selected, it appears pressed in. When a toggle button is deselected, it appears unpressed. In the provided example, a JToggleButton is created with the text "Yes/No" and an initial status of true (selected). The toggle button is added to the JFrame using the FlowLayout layout manager. The user can click on the toggle button to change its state between selected and deselected. When the toggle button is selected, its text color changes to indicate that it is selected.

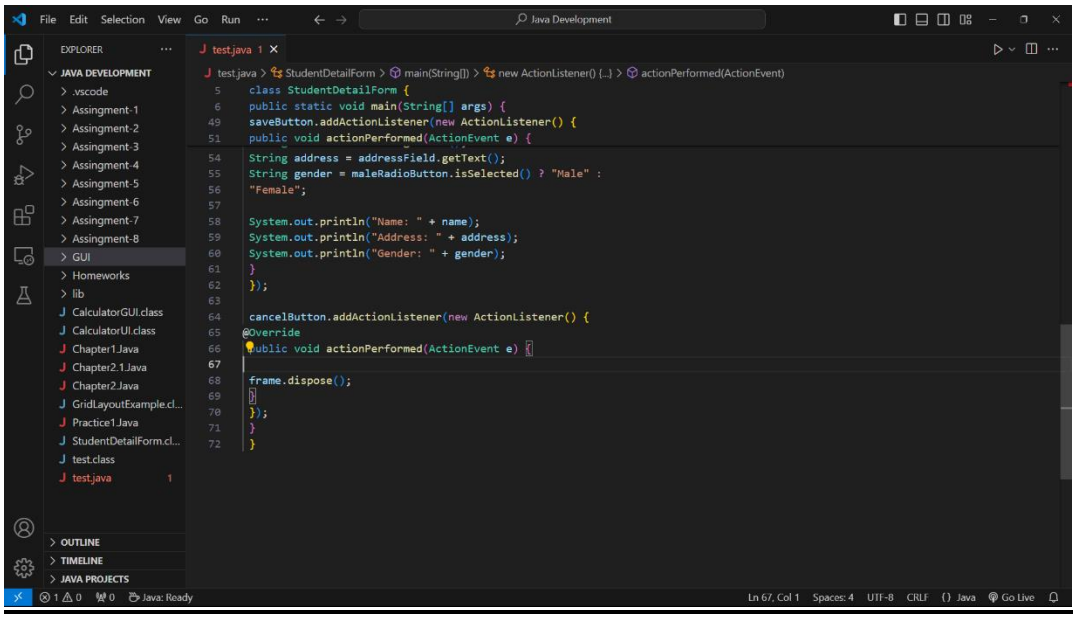
To detect when the state of the toggle button changes, you can add an Action Listener to the togglebutton. The Action Listener' sanction Performed method will be called whenever the state of the togglebutton changes.

Q23: - INPUT: -



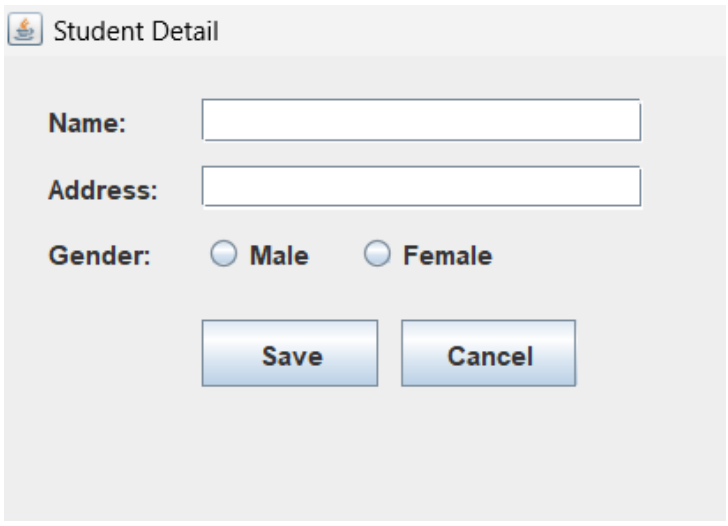


```
class StudentDetailForm {
public static void main(String[] args) {

nameLabel.setBounds(x:20, y:20, width:60, height:20);
nameField.setBounds(x:90, y:20, width:200, height:20);
addressLabel.setBounds(x:20, y:50, width:60, height:20);
addressField.setBounds(x:90, y:50, width:200, height:20);
genderLabel.setBounds(x:20, y:80, width:60, height:20);
maleRadioButton.setBounds(x:90, y:80, width:60, height:20);
femaleRadioButton.setBounds(x:160, y:80, width:80, height:20);
saveButton.setBounds(x:90, y:120, width:80, height:30);
cancelButton.setBounds(x:180, y:120, width:80, height:30);

frame.add(nameLabel);
frame.add(nameField);
frame.add(addressLabel);
frame.add(addressField);
frame.add(genderLabel);
frame.add(maleRadioButton);
frame.add(femaleRadioButton);
frame.add(saveButton);
frame.add(cancelButton);

frame.setVisible(b:true);

saveButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {

String name = nameField.getText();
String address = addressField.getText();
String gender = maleRadioButton.isSelected() ? "Male" :
"Female";

System.out.println("Name: " + name);
System.out.println("Address: " + address);
System.out.println("Gender: " + gender);
}
});
```



```
class StudentDetailForm {
public static void main(String[] args) {
saveButton.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {

String address = addressField.getText();
String gender = maleRadioButton.isSelected() ? "Male" :
"Female";

System.out.println("Name: " + name);
System.out.println("Address: " + address);
System.out.println("Gender: " + gender);
}
});

cancelButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {

frame.dispose();
}
});
}
}
```

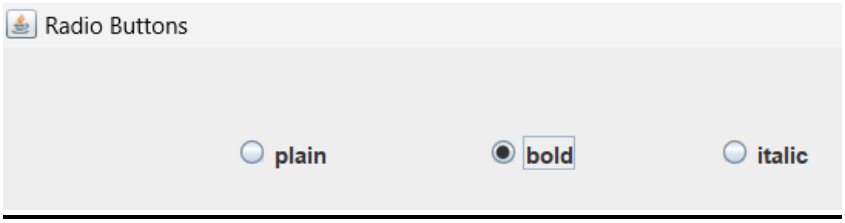
OUTPUT: -



Q24: - INPUT: -

```java
package GUI;

import javax.swing.*;

class RadioButtonExample {
    Run | Debug
    public static void main(String[] args) {
        // Create a new JFrame container
        JFrame frame = new JFrame(title:"Radio Buttons");
        frame.setLayout(manager:null);
        frame.setBounds(x:100, y:100, width:600, height:100);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JRadioButton jRadioButton1 = new JRadioButton(text:"plain");
        JRadioButton jRadioButton2 = new JRadioButton(text:"bold");
        JRadioButton jRadioButton3 = new JRadioButton(text:"italic");
        JRadioButton jRadioButton4 = new JRadioButton(text:"Bolditallic");
        jRadioButton1.setBounds(x:120, y:30, width:120, height:50);
        jRadioButton2.setBounds(x:250, y:30, width:80, height:50);
        jRadioButton3.setBounds(x:370, y:30, width:80, height:50);
        jRadioButton4.setBounds(x:250, y:30, width:80, height:50);

        ButtonGroup buttonGroup = new ButtonGroup();
        buttonGroup.add(jRadioButton1);
        buttonGroup.add(jRadioButton2);
        buttonGroup.add(jRadioButton3);
        buttonGroup.add(jRadioButton4);

        frame.add(jRadioButton1);
        frame.add(jRadioButton2);
        frame.add(jRadioButton3);
        frame.add(jRadioButton4);

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

OUTPUT: -



Q28: - INPUT: -

```java
package GUI;
 import java.awt.FlowLayout;
 import java.awt.event.ActionEvent;
 import java.awt.event.ActionListener;
 import javax.swing.ButtonGroup;
 import javax.swing.JFrame;
 import javax.swing.JRadioButton;
 class RadioButtonDemo extends JFrame implements ActionListener {
 private static final long serialVersionUID = 1L;
 JRadioButton rb1, rb2, rb3, rb4, rb5;
 RadioButtonDemo() {
 super(title:"Radio Button Demo");
setLayout(new FlowLayout(FlowLayout.LEFT, hgap:50, vgap:0));

 rb1 = new JRadioButton(text:"Bird");
 rb2 = new JRadioButton(text:"Cat");
 rb3 = new JRadioButton(text:"Dog");
 rb4 = new JRadioButton(text:"Rabbit");
 rb5 = new JRadioButton(text:"Pig");
 ButtonGroup bg = new ButtonGroup();
 bg.add(rb1);
 bg.add(rb2);
 bg.add(rb3);
 bg.add(rb4);
 bg.add(rb5);
 add(rb1);
 add(rb2);
 add(rb3);
 add(rb4);
 add(rb5);
rb1.addActionListener(this);
 rb2.addActionListener(this);
```

```java
class RadioButtonDemo extends JFrame implements ActionListener {
RadioButtonDemo() {
bg.add(rb1);
bg.add(rb2);
bg.add(rb3);
bg.add(rb4);
bg.add(rb5);
add(rb1);
add(rb2);
add(rb3);
add(rb4);
add(rb5);
rb1.addActionListener(this);
rb2.addActionListener(this);
rb3.addActionListener(this);
rb4.addActionListener(this);
rb5.addActionListener(this);
setVisible(b:true);
setSize(width:200, height:180);
}
public void actionPerformed(ActionEvent e) {
JRadioButton jab = (JRadioButton) e.getSource();
String text = jab.getText();
System.out.println("Selected: " + text);
}
Run | Debug
public static void main(String[] args) {
new RadioButtonDemo();
}
}
```

OUTPUT: -

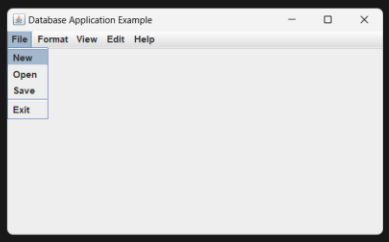
Q30: - INPUT: -

```java
package GUI;

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

class MenuBarToolBar extends JFrame implements ActionListener {
    private JMenuBar menuBar;
    private JMenu fileMenu, formatMenu, viewMenu, editMenu, helpMenu;
    private JMenuItem newMenuItem, openMenuItem, saveMenuItem, exitMenuItem,
            formatItem, viewItem1, viewItem2;
    private JToolBar toolBar;
    private JButton newButton, openButton, saveButton;

    public MenuBarToolBar() {
        super(title:"Database Application Example");

        menuBar = new JMenuBar();

        fileMenu = new JMenu(s:"File");
        formatMenu = new JMenu(s:"Format");
        viewMenu = new JMenu(s:"View");
        editMenu = new JMenu(s:"Edit");
        helpMenu = new JMenu(s:"Help");

        newMenuItem = new JMenuItem(text:"New");
        openMenuItem = new JMenuItem(text:"Open");
        saveMenuItem = new JMenuItem(text:"Save");
        exitMenuItem = new JMenuItem(text:"Exit");

        newMenuItem.addActionListener(this);
        openMenuItem.addActionListener(this);
        saveMenuItem.addActionListener(this);
        exitMenuItem.addActionListener(this);

        fileMenu.add(newMenuItem);
        fileMenu.add(openMenuItem);
        fileMenu.add(saveMenuItem);
        fileMenu.addSeparator();
        fileMenu.add(exitMenuItem);
```

```java
class MenuBarToolBar extends JFrame implements ActionListener {
    public MenuBarToolBar() {

        newMenuItem = new JMenuItem(text:"New");
        openMenuItem = new JMenuItem(text:"Open");
        saveMenuItem = new JMenuItem(text:"Save");
        exitMenuItem = new JMenuItem(text:"Exit");

        newMenuItem.addActionListener(this);
        openMenuItem.addActionListener(this);
        saveMenuItem.addActionListener(this);
        exitMenuItem.addActionListener(this);

        fileMenu.add(newMenuItem);
        fileMenu.add(openMenuItem);
        fileMenu.add(saveMenuItem);
        fileMenu.addSeparator();
        fileMenu.add(exitMenuItem);

        formatItem = new JMenuItem(text:"Formati");
        formatMenu.add(formatItem);

        viewItem1 = new JMenuItem(text:"View1");
        viewItem2 = new JMenuItem(text:"View2");
        viewMenu.add(viewItem1);
        viewMenu.add(viewItem2);

        menuBar.add(fileMenu);
        menuBar.add(formatMenu);
        menuBar.add(viewMenu);
        menuBar.add(editMenu);
        menuBar.add(helpMenu);

        toolBar = new JToolBar();

        newButton = new JButton(new ImageIcon(filename:"[Image of New Icon]"));

        openButton = new JButton(new ImageIcon(filename:"[Image of Open Icon]"));
```

OUTPUT: -

```
PS C:\Users\KIIT\Pictures\Screenshots\Java Development> cd "c:\Users\KIIT\Pictures\Screenshots\Java Development\" ; if ($?) { javac test.java } ;
 if ($?) { java test }
PS C:\Users\KIIT\Pictures\Screenshots\Java Development> Java test.java
Menu item clicked: New
Menu item clicked: Open
Menu item clicked: Save
PS C:\Users\KIIT\Pictures\Screenshots\Java Development> Java test.Java
```

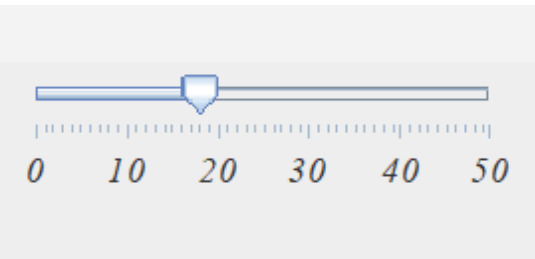


Q31: - INPUT: -

```java
import javax.swing.*;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
import java.awt.*;
class SliderDemo extends JFrame implements ChangeListener {
private static final int FPS_MIN = 0;
private static final int FPS_MAX = 50;
private static final int FPS_INIT = 15;
private Timer timer;
private boolean frozen = false;
private int delay = 1000 / FPS_INIT;
public SliderDemo() {
setTitle(title:"Slider Example");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new FlowLayout());
JSlider framesPerSecond = new JSlider(JSlider.HORIZONTAL, FPS_MIN,
FPS_MAX, FPS_INIT);
framesPerSecond.addChangeListener(this);

framesPerSecond.setMajorTickSpacing(n:10);
framesPerSecond.setMinorTickSpacing(n:1);
framesPerSecond.setPaintTicks(b:true);
framesPerSecond.setPaintLabels(b:true);

Font font = new Font(name:"Serif", Font.ITALIC, size:15);
framesPerSecond.setFont(font);
add(framesPerSecond);

timer = new Timer(delay, e-> {

});
timer.start();
pack();
setLocationRelativeTo(c:null);
}
public void stateChanged(ChangeEvent e) {
JSlider source = (JSlider) e.getSource();
if (!source.getValueIsAdjusting()) {
int fps = (int) source.getValue();
```

```java
class SliderDemo extends JFrame implements ChangeListener {
public void stateChanged(ChangeEvent e) {
if (fps == 0) {
if (!frozen) stopAnimation();
} else {
delay = 1000 / fps;
timer.setDelay(delay);
timer.setInitialDelay(delay * 10);
if (frozen) startAnimation();
}
}
}
private void startAnimation() {

frozen = false;
}
ivate void stopAnimation() {

frozen = true;
}
Run | Debug
public static void main(String[] args) {
SwingUtilities.invokeLater(()-> new
SliderDemo().setVisible(b:true));
}
}
```

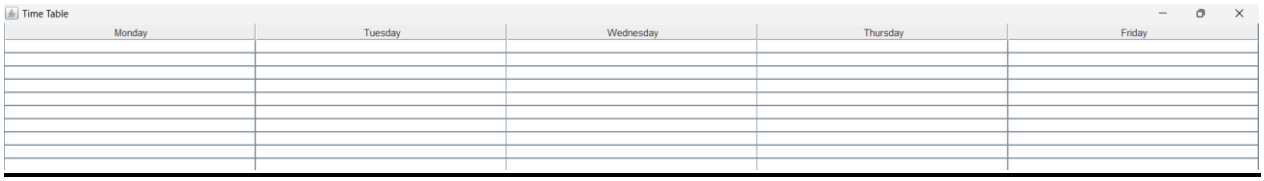
OUTPUT: -



Q33: - INPUT

```java
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

class TimeTableExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame(title:"Time Table");
        DefaultTableModel model = new DefaultTableModel(
                new Object[] { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday" }, rowCount:0);

        for (int i = 0; i <= 9; i++) {
            model.addRow(new Object[] { null, null, null, null });
        }

        JTable table = new JTable(model);
        JScrollPane scrollPane = new JScrollPane(table);
        frame.add(scrollPane);
        frame.setSize(width:500, height:220);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(b:true);
    }
}
```

OUTPUT: -

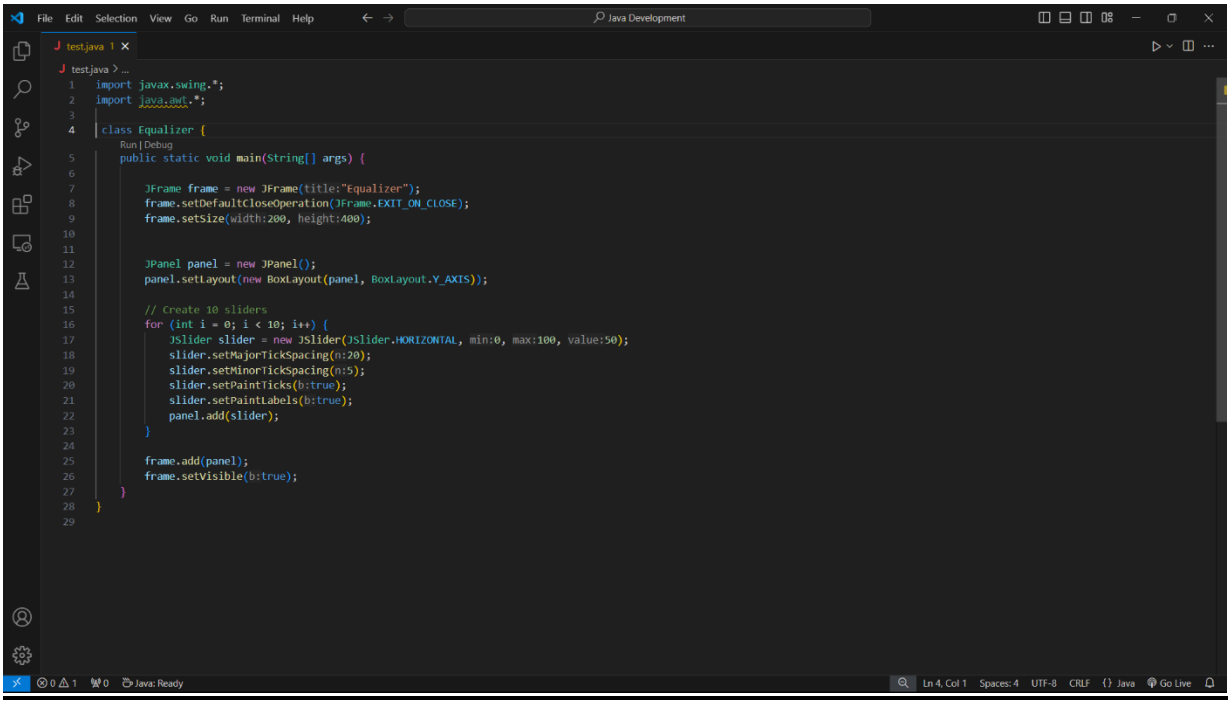


Q25: - INPUT

```java
class ComboBoxDemo {
    public static void main(String[] args) {
        // Create and set up the window.
        JFrame frame = new JFrame(title:"ComboBox Demo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Create and set up the content pane.
        JPanel panel = new JPanel();
        frame.add(panel);
        panel.setLayout(new FlowLayout());

        // Create the combo boxes
        String[] days = new String[31];
        for (int i = 1; i <= 31; i++) {
            days[i - 1] = Integer.toString(i);
        }
        panel.add(new JLabel(text:"Day:"));
        JComboBox<String> dayBox = new JComboBox<>(days);
        panel.add(dayBox);

        String[] months = {"January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"};
        panel.add(new JLabel(text:"Month:"));
        JComboBox<String> monthBox = new JComboBox<>(months);
        panel.add(monthBox);

        String[] years = new String[31];
        for (int i = 1990; i <= 2020; i++) {
            years[i - 1990] = Integer.toString(i);
        }
        panel.add(new JLabel(text:"Year:"));
        JComboBox<String> yearBox = new JComboBox<>(years);
        panel.add(yearBox);

        //Display the window.
        frame.pack();
        frame.setVisible(b:true);
    }
}
```

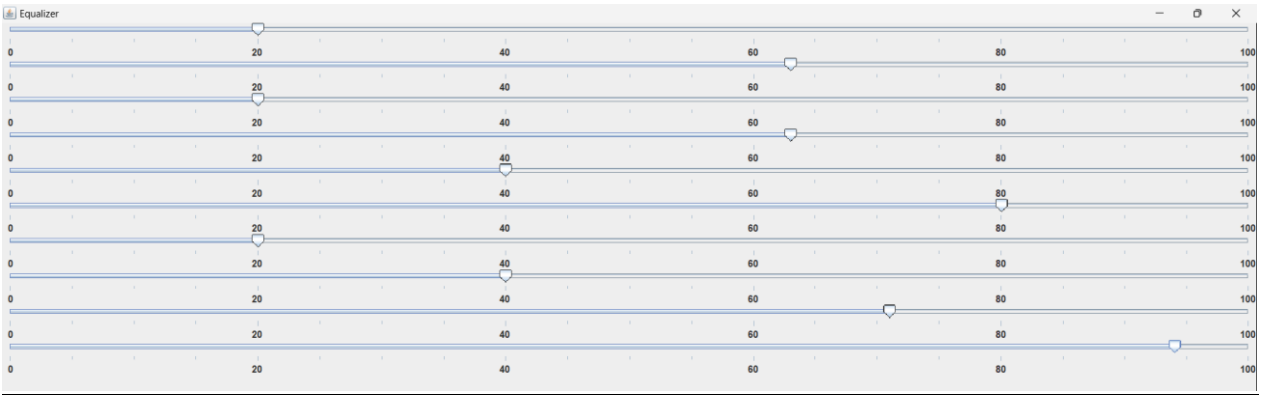
OUTPUT: -

Q26: - INPUT

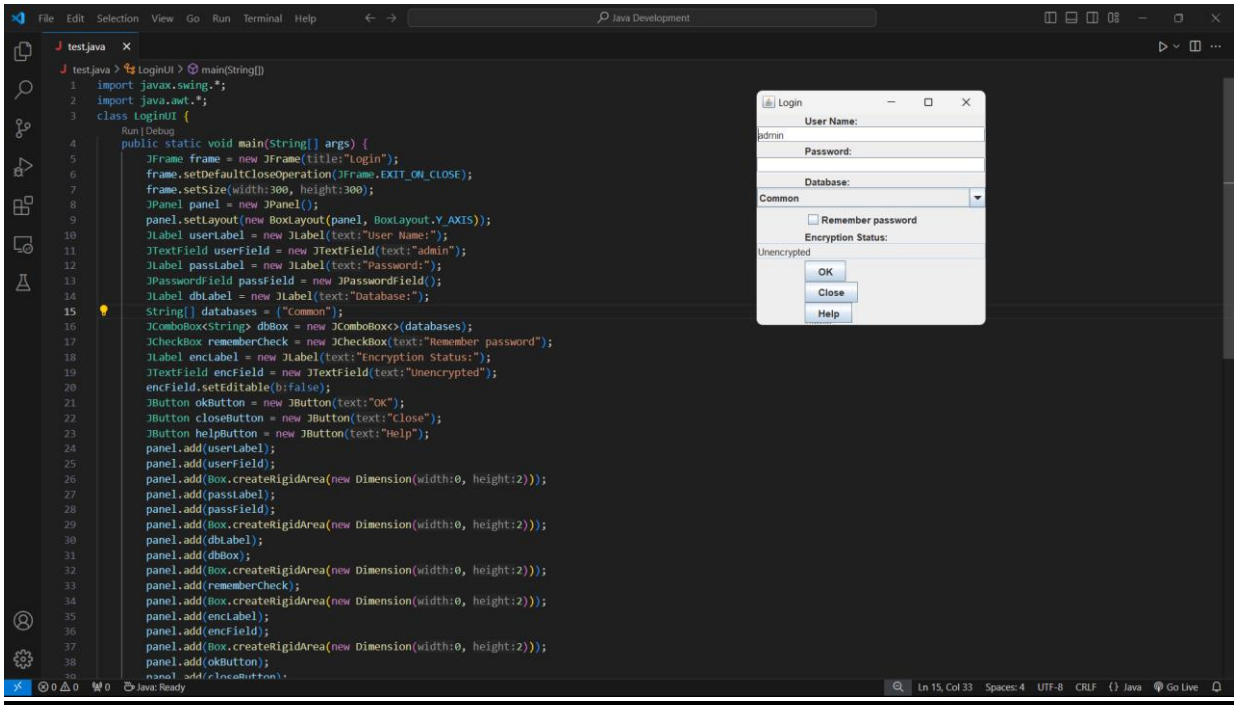


```java
import javax.swing.*;
import java.awt.*;

class Equalizer {
    Run | Debug
    public static void main(String[] args) {

        JFrame frame = new JFrame(title:"Equalizer");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(width:200, height:400);


        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

        // Create 10 sliders
        for (int i = 0; i < 10; i++) {
            JSlider slider = new JSlider(JSlider.HORIZONTAL, min:0, max:100, value:50);
            slider.setMajorTickSpacing(n:20);
            slider.setMinorTickSpacing(n:5);
            slider.setPaintTicks(b:true);
            slider.setPaintLabels(b:true);
            panel.add(slider);
        }

        frame.add(panel);
        frame.setVisible(b:true);
    }
}
```

OUTPUT: -



Q27: - INPUT/OUTPUT



****