

6. Swing

- a) Create a swing application that randomly changes color on button click.

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

public class ChangeColor extends JFrame
{
    private JPanel colorPanel;
    private JButton changeColorButton;

    public ChangeColor()
    {
        setTitle("Random Color Changer");
        setSize(300, 200);
```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new BorderLayout( ));

colorPanel = new JPanel( );
changeColorButton = new JButton("Change Color");

add(colorPanel, BorderLayout.CENTER);
add(changeColorButton, BorderLayout.SOUTH);

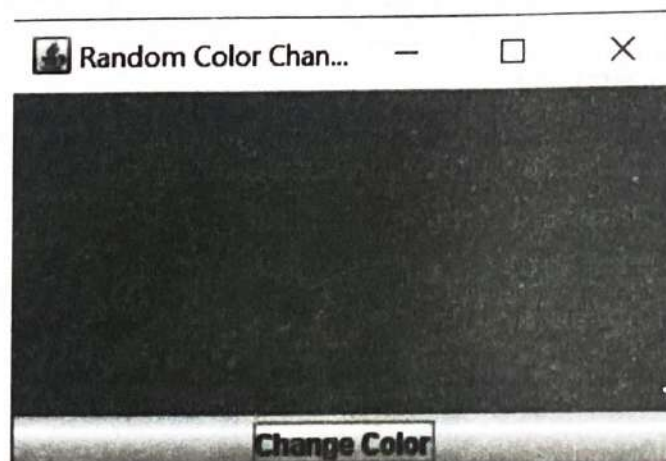
changeColorButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        changeColor();
    }
});
}

private void changeColor( ) {
    Random random = new Random();
    Color randomColor = new Color(random.nextInt(256), random.nextInt(256),
                                   random.nextInt(256));

    colorPanel.setBackground(randomColor);
}

public static void main(String[ ] args) {
    SwingUtilities.invokeLater(() -> {
        ChangeColor app = new ChangeColor( );
        app.setVisible(true);
    });
}
}
```

Output :



- b) Create a Swing application to demonstrate use of `TextArea` using `scrollpane` to show content of text file in `textarea` selected using `file chooser`.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class ShowFileInTextArea extends JFrame
{
    private JTextArea textArea = new JTextArea(20, 40);
    private JButton openFileButton = new JButton("Open File");

    public ShowFileInTextArea()
    {
        setTitle("File Viewer");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        JScrollPane scrollPane = new JScrollPane(textArea);
        add(scrollPane, BorderLayout.CENTER);
        add(openFileButton, BorderLayout.SOUTH);
        openFileButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                openFile();
            }
        });
        pack();
        setLocationRelativeTo(null); // Center the frame on the screen
    }

    private void openFile() {
        JFileChooser fileChooser = new JFileChooser();
        int result = fileChooser.showOpenDialog(this);
        if (result == JFileChooser.APPROVE_OPTION) {
            try (BufferedReader reader = new BufferedReader
                (new FileReader(fileChooser.getSelectedFile( ))) {
                StringBuilder content = new StringBuilder();
                String line;
                while ((line = reader.readLine( )) != null) {
                    content.append(line).append("\n");
                }
            }
        }
    }
}
```

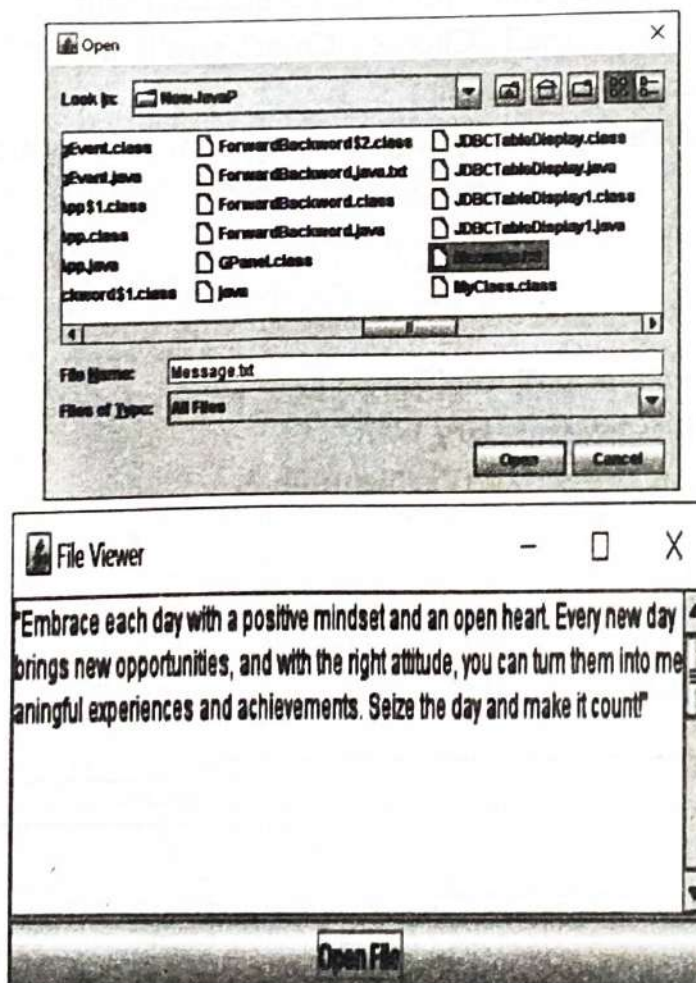
```

        textArea.setLineWrap(true);
        textArea.setText(content.toString());
    } catch (IOException e) {
        JOptionPane.showMessageDialog(this, "Error reading the file.",
            "Error", JOptionPane.ERROR_MESSAGE);
    }
}
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        new ShowFileInTextArea().setVisible(true);
    });
}
}

```

Output :



- c) Create a Swing application to demonstrate use of scrollpane to change its color selected using colour chooser.

```

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

```

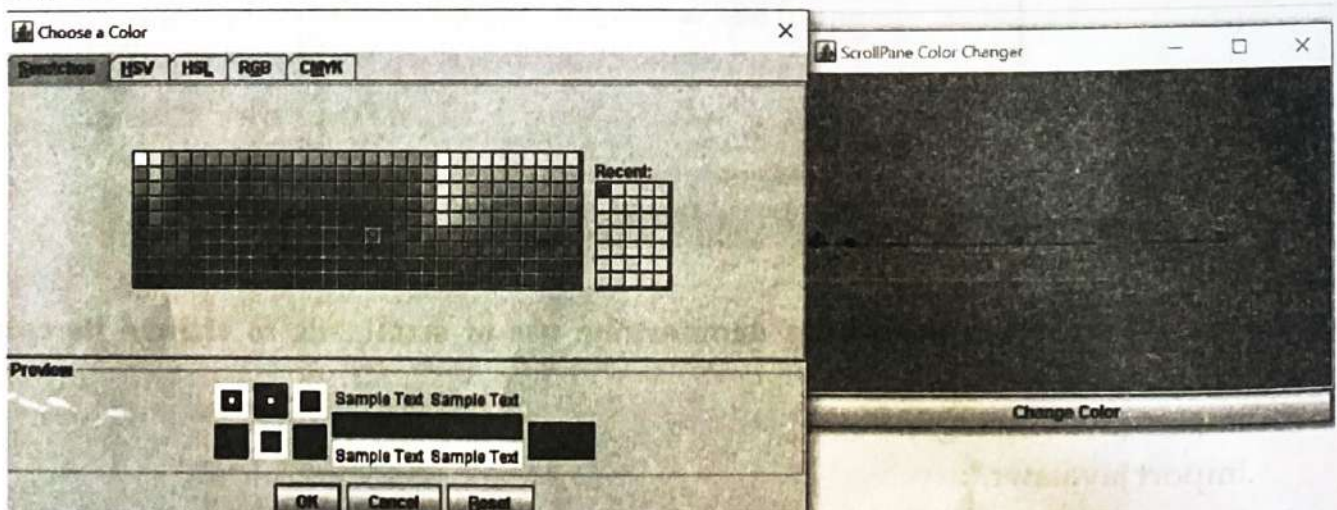


```

public class ColorChnager extends JFrame
{
    private JScrollPane scrollPane = new JScrollPane( );
    private JButton changeColorButton = new JButton("Change Color");
    public ColorChnager( ) {
        setTitle("ScrollPane Color Changer");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout( ));
        add(scrollPane, BorderLayout.CENTER);
        add(changeColorButton, BorderLayout.SOUTH);
        changeColorButton.addActionListener(new ActionListener( ) {
            @Override
            public void actionPerformed(ActionEvent e) {
                Color selectedColor = JColorChooser.showDialog
                    (null, "Choose a Color", scrollPane.getBackground( ));
                if (selectedColor != null) {
                    scrollPane.getViewport().setBackground(selectedColor);
                }
            }
        });
    }
    public static void main(String[ ] args) {
        SwingUtilities.invokeLater(() -> {
            new ColorChnager().setVisible(true);
        });
    }
}

```

Output :



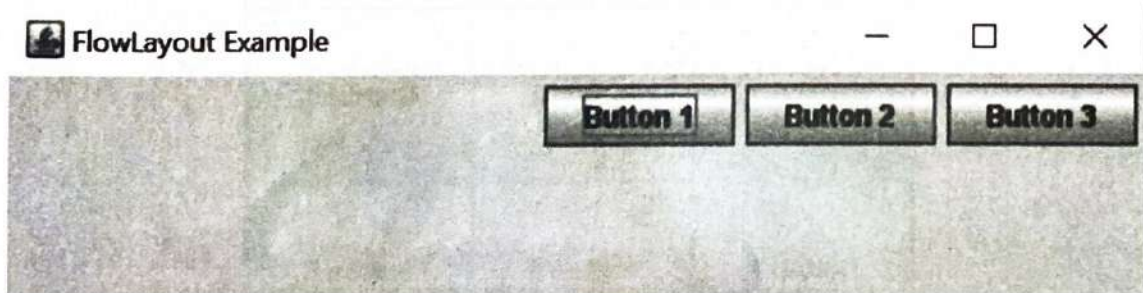
7. Layouts : Write programs for the following layouts :

a) Flow Layout Example.

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

```
public class DemoFlowLayout  
{  
    public static void main(String[ ] args)  
    {  
        // Create a JFrame (main window)  
        JFrame frame = new JFrame("FlowLayout Example");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(300, 100);  
  
        // Create a JPanel with FlowLayout  
        JPanel panel = new JPanel(new FlowLayout(FlowLayout.RIGHT));  
  
        // Create three buttons  
        JButton button1 = new JButton("Button 1");  
        JButton button2 = new JButton("Button 2");  
        JButton button3 = new JButton("Button 3");  
  
        // Add the buttons to the panel  
        panel.add(button1);  
        panel.add(button2);  
        panel.add(button3);  
  
        frame.add(panel);    // Add the panel to the frame  
        frame.setVisible(true);    // Make the JFrame visible  
    }  
}
```

Output :



b) Grid Layout Example.

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



```

public class DemoGridLayout {
    public static void main(String[] args) {
        // Create a JFrame (main window)
        JFrame frame = new JFrame("GridLayout Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);

        // Create a JPanel with GridLayout (2 rows, 3 columns)
        JPanel panel = new JPanel(new GridLayout(2, 3));

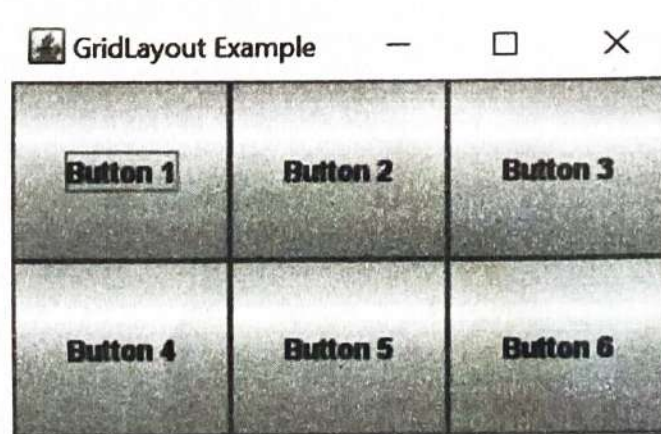
        // Create buttons to be placed in the grid
        JButton button1 = new JButton("Button 1");
        JButton button2 = new JButton("Button 2");
        JButton button3 = new JButton("Button 3");
        JButton button4 = new JButton("Button 4");
        JButton button5 = new JButton("Button 5");
        JButton button6 = new JButton("Button 6");

        // Add buttons to the panel (they will be arranged in a 2x3 grid)
        panel.add(button1);
        panel.add(button2);
        panel.add(button3);
        panel.add(button4);
        panel.add(button5);
        panel.add(button6);

        frame.add(panel);           // Add the panel to the frame
        frame.setVisible(true);    // Make the JFrame visible
    }
}

```

Output :



c) Border Layout Example.

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
public class DemoBorderLayout {  
    public static void main(String[] args) {  
        // Create a JFrame (main window)  
        JFrame frame = new JFrame("BorderLayout Example");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(300, 200);  
  
        // Create buttons for each region (North, South, East, West, Center)  
        JButton northButton = new JButton("North");  
        JButton southButton = new JButton("South");  
        JButton eastButton = new JButton("East");  
        JButton westButton = new JButton("West");  
        JButton centerButton = new JButton("Center");  
  
        // Set up BorderLayout for the frame's content pane  
        Container contentPane = frame.getContentPane();  
        contentPane.setLayout(new BorderLayout());  
  
        // Add buttons to their respective regions  
        contentPane.add(northButton, BorderLayout.NORTH);  
        contentPane.add(southButton, BorderLayout.SOUTH);  
        contentPane.add(eastButton, BorderLayout.EAST);  
        contentPane.add(westButton, BorderLayout.WEST);  
        contentPane.add(centerButton, BorderLayout.CENTER);  
  
        frame.setVisible(true);    // Make the JFrame visible  
    }  
}
```

Output :

