

## **Advance Java Seminar**

### **Programs**

#### **1. Program to create frame in AWT**

```
import java.awt.Frame;

public class SimpleFrameExample {
    public static void main(String[] args) {
        // Create a frame
        Frame frame = new Frame("Simple Frame Example");

        // Set the size of the frame
        frame.setSize(300, 200);

        // Make the frame visible
        frame.setVisible(true);
    }
}
```

#### **2. Adding controls to frame**

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

public class BasicControlsDemo {
    // Declare AWT components
    private Frame frame;
    private Label label;
    private Button button;
    private TextField textField;
    private TextArea textArea;
    private Checkbox checkbox;
    private List list;

    // Constructor to initialize components
    public BasicControlsDemo() {
        frame = new Frame("Basic Controls Demo");
        label = new Label("Label");
        button = new Button("Click Me");
        textField = new TextField();
        textArea = new TextArea();
        checkbox = new Checkbox("Check Me");
        list = new List();

        // Set layout manager for the frame
    }
}
```

```

frame.setLayout(new FlowLayout());

// Add components to the frame
frame.add(label);
frame.add(button);
frame.add(textField);
frame.add(textArea);
frame.add(checkbox);
frame.add(list);

// Set action listener for the button
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("Button clicked!");
    }
});

// Set window closing listener
frame.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});

// Set frame properties
frame.setSize(400, 300);
frame.setVisible(true);
}

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

```

### 3. creating simple applet

```

import java.applet.Applet;
import java.awt.Graphics;

public class HelloWorldApplet extends Applet {

    public void paint(Graphics g) {
        g.drawString("Hello, World!", 20, 20);
    }
}

Html added

<html>
<head>

```

```
<title>Hello World Applet</title>
</head>
<body>
<applet code="HelloWorldApplet.class" width="300" height="200">
</applet>
</body>
</html>
```

#### 4. adding controls to applet

```
import java.applet.Applet;
import java.awt.Button;
import java.awt.Label;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class LabelButtonApplet extends Applet implements ActionListener {

    private Label label;
    private Button button;

    public void init() {
        label = new Label("Click the button!");
        button = new Button("Click Me");

        // Add action listener to the button
        button.addActionListener(this);

        // Add components to the applet
        add(label);
        add(button);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == button) {
            label.setText("Button Clicked!");
        }
    }
}
```

#### 5. flow layout

```
import java.awt.Button;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
```

```

public class FlowLayoutExample {

    public static void main(String[] args) {
        Frame frame = new Frame("FlowLayout Example");

        // Create components
        Label label = new Label("Label");
        Button button = new Button("Button");

        // Set layout manager for the frame
        frame.setLayout(new FlowLayout());

        // Add components to the frame
        frame.add(label);
        frame.add(button);

        // Set window closing listener
        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });

        // Set frame properties
        frame.setSize(300, 150);
        frame.setVisible(true);
    }
}

```

## 6. Border layout

```

import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.Frame;

public class BorderLayoutExample {

    public static void main(String[] args) {
        Frame frame = new Frame("BorderLayout Example");

        // Create buttons
        Button button1 = new Button("Button 1");
        Button button2 = new Button("Button 2");
        Button button3 = new Button("Button 3");
        Button button4 = new Button("Button 4");
        Button button5 = new Button("Button 5");

        // Add buttons to the frame with BorderLayout
        frame.add(button1, BorderLayout.NORTH);
        frame.add(button2, BorderLayout.SOUTH);
    }
}

```

```

        frame.add(button3, BorderLayout.WEST);
        frame.add(button4, BorderLayout.EAST);
        frame.add(button5, BorderLayout.CENTER);

    // Set frame properties
    frame.setSize(300, 200);
    frame.setVisible(true);
}
}

```

## 7. Grid layout

```

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

public class ButtonGridLayoutExample {
    private Frame frame;

    public ButtonGridLayoutExample() {
        frame = new Frame("Button Grid Layout Example");

        // Create buttons
        Button button1 = new Button("Button 1");
        Button button2 = new Button("Button 2");
        Button button3 = new Button("Button 3");
        Button button4 = new Button("Button 4");
        Button button5 = new Button("Button 5");

        // Create GridLayout with 2 rows and 3 columns
        GridLayout gridLayout = new GridLayout(2, 3);

        // Set GridLayout for the frame
        frame.setLayout(gridLayout);

        // Add buttons to the frame
        frame.add(button1);
        frame.add(button2);
        frame.add(button3);
        frame.add(button4);
        frame.add(button5);

        // Set window closing listener
        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }
}

```

```

// Set frame properties
frame.setSize(300, 200);
frame.setVisible(true);
}

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

```

## 8. Grid bag layout

```

import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import javax.swing.JButton;
import javax.swing.JFrame;

public class GridBagLayoutExample {

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

        // Create a GridBagLayout manager
        GridBagLayout layout = new GridBagLayout();
        frame.setLayout(layout);

        // Create buttons
        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");

        // Create GridBagConstraints to control the layout
        GridBagConstraints gbc = new GridBagConstraints();

        // Set constraints for button1
        gbc.gridx = 0;
        gbc.gridy = 0;
        frame.add(button1, gbc);

        // Set constraints for button2
        gbc.gridx = 1;
        gbc.gridy = 0;
        frame.add(button2, gbc);

        // Set constraints for button3
        gbc.gridx = 2;
        gbc.gridy = 0;
    }
}

```

```

frame.add(button3, gbc);

// Set constraints for button4
gbc.gridx = 0;
gbc.gridy = 1;
frame.add(button4, gbc);

// Set constraints for button5
gbc.gridx = 1;
gbc.gridy = 1;
frame.add(button5, gbc);

// Set frame properties
frame.setSize(300, 200);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
}
}

```

## 9. Card layout

```

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

public class CardLayoutExample extends Frame implements ActionListener {

    private CardLayout cardLayout;
    private Panel cardPanel;

    public CardLayoutExample() {
        cardLayout = new CardLayout();
        cardPanel = new Panel();

        setLayout(new BorderLayout());
        cardPanel.setLayout(cardLayout);

        // Create buttons for different "cards"
        Button card1Button = new Button("Card 1");
        Button card2Button = new Button("Card 2");
        Button card3Button = new Button("Card 3");

        // Add action listeners to the buttons
        card1Button.addActionListener(this);
        card2Button.addActionListener(this);
        card3Button.addActionListener(this);

        // Add buttons to the card panel
        cardPanel.add(card1Button, "Card 1");
        cardPanel.add(card2Button, "Card 2");
    }
}

```

```

cardPanel.add(card3Button, "Card 3");

// Add the card panel to the frame
add(cardPanel, BorderLayout.CENTER);

// Set the frame properties
setTitle("Card Layout Example");
setSize(300, 200);
setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    cardLayout.next(cardPanel);
}

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

```

## 10. Menu bar

```

import java.awt.*;
import java.awt.event.*;

public classMenuBarExample extends Frame {

    publicMenuBarExample() {
        // Create a MenuBar
        MenuBar menuBar = new MenuBar();

        // Create Menus
        Menu fileMenu = new Menu("File");
        Menu editMenu = new Menu("Edit");
        Menu helpMenu = new Menu("Help");

        // Create MenuItem
        MenuItem newItem = new MenuItem("New");
        MenuItem openMenuItem = new MenuItem("Open");
        MenuItem saveMenuItem = new MenuItem("Save");
        MenuItem exitMenuItem = new MenuItem("Exit");

        MenuItem cutMenuItem = new MenuItem("Cut");
        MenuItem copyMenuItem = new MenuItem("Copy");
        MenuItem pasteMenuItem = new MenuItem("Paste");

        MenuItem aboutMenuItem = new MenuItem("About");

        // Add MenuItem to Menus
        fileMenu.add(newMenuItem);

```

```

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

editMenu.add(cutMenuItem);
editMenu.add(copyMenuItem);
editMenu.add(pasteMenuItem);

helpMenu.add(aboutMenuItem);

// Add Menus to MenuBar
menuBar.add(fileMenu);
menuBar.add(editMenu);
menuBar.add(helpMenu);

// Set MenuBar for the frame
setMenuBar(menuBar);

// Set window closing listener
addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});

// Set frame properties
setTitle("Menu Bar Example");
setSize(300, 200);
setVisible(true);
}

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

```

## 11. File dialog

```

import java.awt.*;
import java.awt.event.*;

public class FileDialogExample extends Frame {

    public FileDialogExample() {
        setTitle("File Dialog Example");

        // Create a FileDialog with the specified title and mode
        FileDialog fileDialog = new FileDialog(this, "Open File", FileDialog.LOAD);
    }
}

```

```

// Create a button to trigger the FileDialog
Button openButton = new Button("Open File");

// Add an action listener to the button
openButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Display the FileDialog when the button is clicked
        fileDialog.setVisible(true);

        // Get the selected file and display its path
        String directory = fileDialog.getDirectory();
        String file = fileDialog.getFile();
        if (file != null) {
            System.out.println("Selected File: " + directory + file);
        }
    }
});

// Set layout manager for the frame
setLayout(new FlowLayout());

// Add components to the frame
add(openButton);

// Set window closing listener
addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});

// Set frame properties
setSize(300, 200);
setVisible(true);
}

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

```

## 12. Image Icon class

```

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

public class ImageIconExample {

    public ImageIconExample() {
        // Create a JFrame

```

```

JFrame frame = new JFrame("ImageIcon Example");

// Create an ImageIcon with the specified image file path
ImageIcon icon = new ImageIcon("path/to/your/image.jpg");

// Create a JLabel with the ImageIcon
JLabel label = new JLabel(icon);

// Set layout manager for the frame
frame.setLayout(new FlowLayout());

// Add the label to the frame
frame.add(label);

// Set default close operation
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Set frame properties
frame.setSize(300, 200);
frame.setVisible(true);
}

public static void main(String[] args) {
    // Ensure to replace "path/to/your/image.jpg" with the actual path to your image
file
    new ImageIconExample();
}
}

```

### 13. Tabbed panes

```

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

public class TabbedPaneExample extends JFrame {

    public TabbedPaneExample() {
        setTitle("TabbedPane Example");

        // Create a JTabbedPane
        JTabbedPane tabbedPane = new JTabbedPane();

        // Create panels for each tab
        JPanel panel1 = new JPanel();
        JPanel panel2 = new JPanel();
        JPanel panel3 = new JPanel();

        // Add components to the panels
        panel1.add(new JLabel("Content for Tab 1"));
        panel2.add(new JLabel("Content for Tab 2"));
    }
}

```

```

panel3.add(new JLabel("Content for Tab 3"));

// Add tabs to the tabbedPane
tabbedPane.addTab("Tab 1", panel1);
tabbedPane.addTab("Tab 2", panel2);
tabbedPane.addTab("Tab 3", panel3);

// Set layout manager for the frame
setLayout(new BorderLayout());

// Add the tabbedPane to the frame
add(tabbedPane, BorderLayout.CENTER);

// Set default close operation
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Set frame properties
setSize(400, 300);
 setLocationRelativeTo(null); // Center the frame
setVisible(true);
}

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

```

## 14. trees

```

import javax.swing.*;
import javax.swing.tree.DefaultMutableTreeNode;

public class SwingTreeExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Swing Tree Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create a root node
        DefaultMutableTreeNode rootNode = new DefaultMutableTreeNode("Root");

        // Create child nodes
        DefaultMutableTreeNode node1 = new DefaultMutableTreeNode("Node 1");
        DefaultMutableTreeNode node2 = new DefaultMutableTreeNode("Node 2");

        // Add child nodes to the root node
        rootNode.add(node1);
        rootNode.add(node2);

        // Create the tree with the root node

```

```

JTree tree = new JTree(rootNode);

// Create a scroll pane to hold the tree
JScrollPane scrollPane = new JScrollPane(tree);

// Add the scroll pane to the frame
frame.add(scrollPane);

// Set frame properties
frame.setSize(300, 200);
frame.setLocationRelativeTo(null);
frame.setVisible(true);
};

}
}

```

## 15. tables

```

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

public class testing extends JFrame {

    public testing() {
        setTitle("Table Example");

        // Create column names and data for the table
        String[] columnNames = {"Name", "Age", "Occupation"};
        Object[][] data = {
            {"John Doe", 30, "Engineer"},
            {"Jane Smith", 25, "Teacher"},
            {"Bob Johnson", 40, "Doctor"},
            {"Alice Williams", 35, "Artist"}
        };

        // Create a JTable with the given data and column names
        JTable table = new JTable(data, columnNames);

        // Set the layout manager for the frame
        setLayout(new BorderLayout());

        // Add the table to a JScrollPane to make it scrollable
        JScrollPane scrollPane = new JScrollPane(table);

        // Add the scroll pane to the frame
        add(scrollPane, BorderLayout.CENTER);

        // Set window properties
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setVisible(true);
    }

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

```

## 16. Progress bar

```

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

public class testing extends JFrame {

    private JProgressBar progressBar;
    private JButton startButton;

    public testing() {
        setTitle("Progress Bar Example");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create components
        progressBar = new JProgressBar(0, 100);
        startButton = new JButton("Start");

        // Set layout manager
        setLayout(new FlowLayout());

        // Add components to the frame
        add(progressBar);
        add(startButton);

        // Add action listener to the button
        startButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // Perform a task in a separate thread
                SwingWorker<Void, Integer> worker = new SwingWorker<Void, Integer>() {
                    @Override
                    protected Void doInBackground() throws Exception {
                        for (int i = 0; i <= 100; i++) {
                            Thread.sleep(50); // Simulate a time-consuming task
                            publish(i); // Update the progress
                        }
                        return null;
                    }

                    @Override
                    protected void process(List<Integer> chunks) {
                        // Update the progress bar
                        progressBar.setValue(chunks.get(chunks.size() - 1));
                    }
                };
                // Execute the SwingWorker
                worker.execute();
            }
        });
    }

    // Set frame properties
    setSize(300, 100);
    setLocationRelativeTo(null); // Center the frame on the screen
    setVisible(true);
}

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

```

## 17. mouse event

```

import java.awt.*;
import java.awt.event.*;

```

```
public class MouseEventExample extends Frame {  
  
    private Panel panel;  
  
    public MouseEventExample() {  
        setTitle("Mouse Event Example");  
  
        // Create a panel to handle mouse events  
        panel = new Panel();  
        panel.addMouseListener(new MyMouseListener());  
  
        // Set layout manager for the frame  
        setLayout(new FlowLayout());  
  
        // Add components to the frame  
        add(panel);  
  
        // Set window closing listener  
        addWindowListener(new WindowAdapter() {  
            public void windowClosing(WindowEvent e) {  
                System.exit(0);  
            }  
        });  
  
        // Set frame properties  
        setSize(300, 200);  
        setVisible(true);  
    }  
  
    // Inner class to implement MouseListener  
    class MyMouseListener implements MouseListener {  
        public void mouseClicked(MouseEvent e) {  
            System.out.println("Mouse Clicked at: " + e.getX() + ", " + e.getY());  
        }  
  
        public void mouseEntered(MouseEvent e) {  
            System.out.println("Mouse Entered");  
        }  
  
        public void mouseExited(MouseEvent e) {  
            System.out.println("Mouse Exited");  
        }  
  
        public void mousePressed(MouseEvent e) {  
            System.out.println("Mouse Pressed at: " + e.getX() + ", " + e.getY());  
        }  
  
        public void mouseReleased(MouseEvent e) {  
            System.out.println("Mouse Released at: " + e.getX() + ", " + e.getY());  
        }  
    }  
}
```

```
}

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

## 18. Keyboard event

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

public class KeyEventExample extends Frame implements KeyListener {

    private TextArea textArea;

    public KeyEventExample() {
        setTitle("Key Event Example");

        textArea = new TextArea();
        textArea.addKeyListener(this);

        // Set layout manager for the frame
        setLayout(new BorderLayout());

        // Add components to the frame
        add(textArea, BorderLayout.CENTER);

        // Set window closing listener
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });

        // Set frame properties
        setSize(300, 200);
        setVisible(true);
    }

    // KeyListener methods
    public void keyTyped(KeyEvent e) {
        // Called when a key is typed (pressed and released)
        char keyChar = e.getKeyChar();
        textArea.append("Key Typed: " + keyChar + "\n");
    }

    public void keyPressed(KeyEvent e) {
        // Called when a key is pressed
        int keyCode = e.getKeyCode();
```

```

    textArea.append("Key Pressed: " + KeyEvent.getKeyText(keyCode) + "\n");
}

public void keyReleased(KeyEvent e) {
    // Called when a key is released
    int keyCode = e.getKeyCode();
    textArea.append("Key Released: " + KeyEvent.getKeyText(keyCode) + "\n");
}

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

```

## 19. Adapter class

```

import java.awt.*;
import java.awt.event.*;

// Custom adapter class for mouse events
class MyMouseAdapter extends MouseAdapter {
    public void mouseClicked(MouseEvent e) {
        System.out.println("Mouse Clicked");
    }

    public void mousePressed(MouseEvent e) {
        System.out.println("Mouse Pressed");
    }

    public void mouseReleased(MouseEvent e) {
        System.out.println("Mouse Released");
    }
}

public class MouseEventAdapterExample extends Frame {

    public MouseEventAdapterExample() {
        setTitle("Mouse Event Adapter Example");

        // Create an instance of the custom adapter class
        MyMouseAdapter mouseAdapter = new MyMouseAdapter();

        // Add the adapter to the frame to handle mouse events
        addMouseListener(mouseAdapter);

        // Set layout manager for the frame
        setLayout(new FlowLayout());

        // Set window closing listener
        addWindowListener(new WindowAdapter() {

```

```
public void windowClosing(WindowEvent e) {
    System.exit(0);
}
});

// Set frame properties
setSize(300, 200);
setVisible(true);
}

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

