ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Object Oriented Programming

Lecture 3: Graphics and GUI

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

GUI and Graphics Overview

Basic GUI

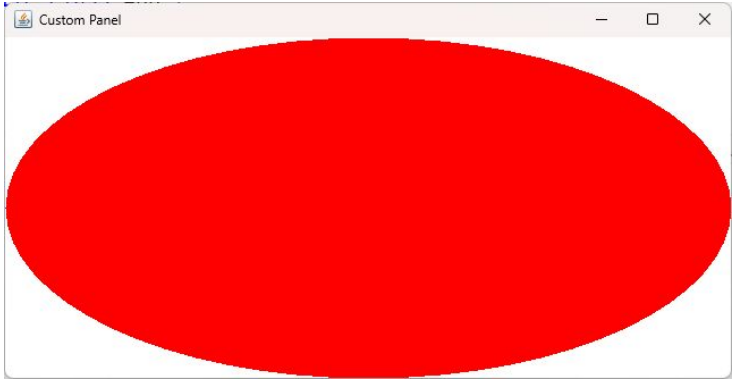
```
import java.awt.*;
import javax.swing.*;

public class App {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Login Panels");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // Set up first subpanel
        JPanel panel = new JPanel();
        panel.setPreferredSize(new Dimension(500, 50));
        panel.add(new JLabel("User Name:"));
        panel.add(new JTextField(10));
        panel.add(new JLabel("Password:"));
        panel.add(new JTextField(10));
        panel.add(new JButton("Login"));
        frame.getContentPane().add(panel);
        frame.pack();
        frame.setVisible(true);
    }
}
```

Basic Graphics

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

public class GraphicsPanel extends JPanel {
    public GraphicsPanel(int width, int height, Color backColor) {
        setPreferredSize(new Dimension(width, height));
        setBackground(backColor);
    }
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.red);
        g.fillOval(0, 0, getWidth(), getHeight());
    }
}
```

Basic Graphics

```
import java.awt.*;
import javax.swing.*;
public class App {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Custom Panel");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        GraphicsPanel panel = new GraphicsPanel(640, 300, Color.white);
        frame.getContentPane().add(panel);
        frame.pack();
        frame.setVisible(true);
    }
}
```

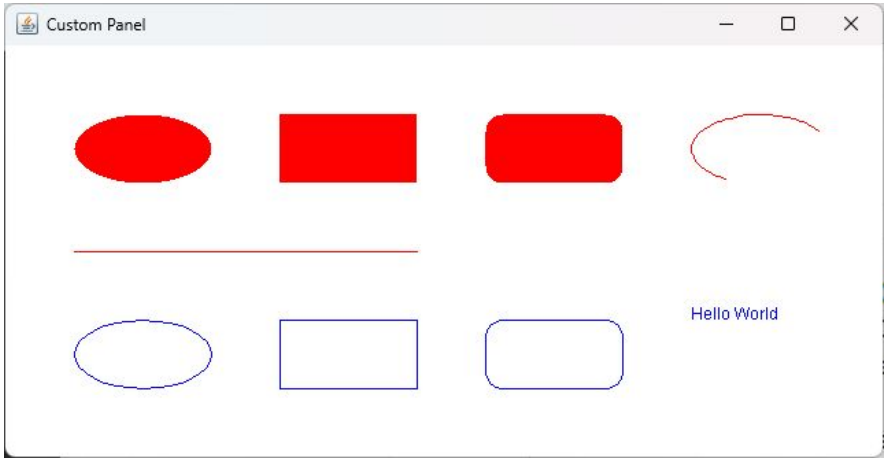


ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Java Graphics

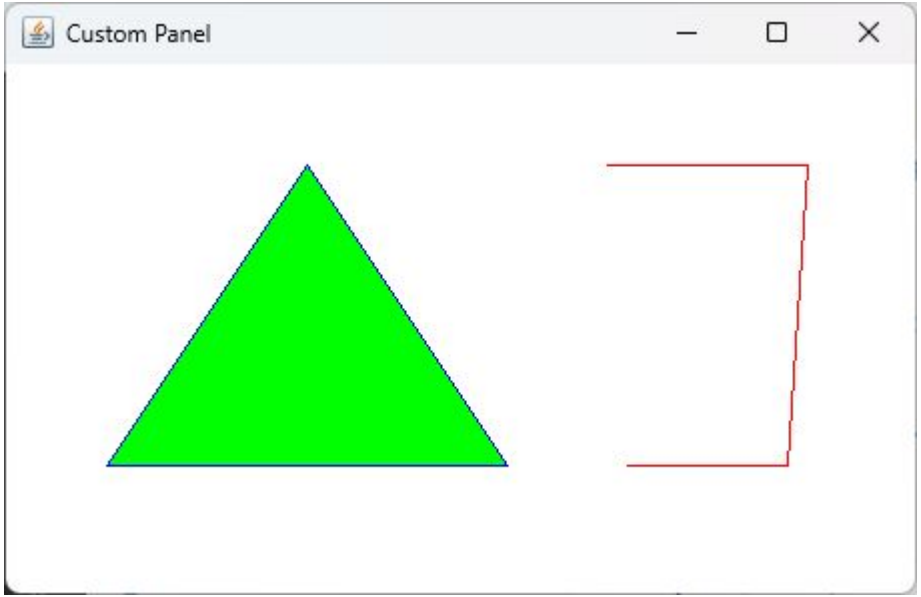
Painting Shapes

```
super.paintComponent(g);
g.setColor(Color.red);
g.fillOval(50, 50, 100, 50);
g.fillRect(200, 50, 100, 50);
g.fillRoundRect(350, 50, 100, 50, 25, 25);
g.drawLine(50, 150, 300, 150);
g.drawArc(500, 50, 100, 50, 30, 210);
g.setColor(Color.blue);
g.drawOval(50, 200, 100, 50);
g.drawRect(200, 200, 100, 50);
g.drawRoundRect(350, 200, 100, 50, 25, 25);
g.drawString("Hello World", 500, 200);
```



Drawing Polygons and Polylines

```
super.paintComponent(g);
Polygon polygon = new Polygon();
polygon.addPoint(150, 50);
polygon.addPoint(250, 200);
polygon.addPoint(50, 200);
g.setColor(Color.green);
g.fillPolygon(polygon);
g.setColor(Color.blue);
g.drawPolygon(polygon);
int x[] = { 300, 400, 390, 310 };
int y[] = { 50, 50, 200, 200 };
g.setColor(Color.red);
g.drawPolyline(x, y, 4);
```



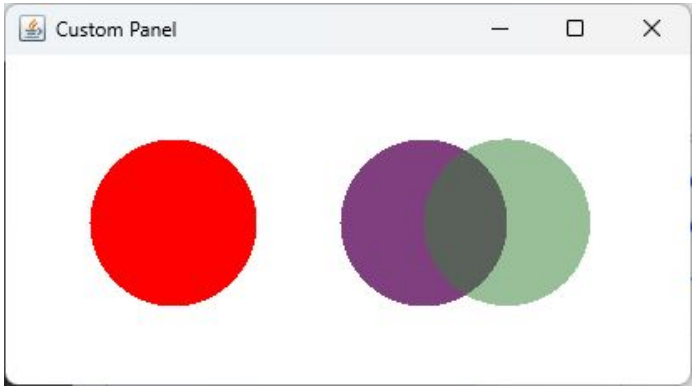
Draw Images using Toolkit

```
public class GraphicsPanel extends JPanel {
    private Image image;
    public GraphicsPanel(int width, int height, Color backColor) {
        setPreferredSize(new Dimension(width, height));
        setBackground(backColor);
        image = Toolkit.getDefaultToolkit().getImage("smile.gif");
    }
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawImage(image, 50, 50, this);
        g.drawImage(image, 250, 50, 200, 100, this);
    }
}
```

Using Java Color

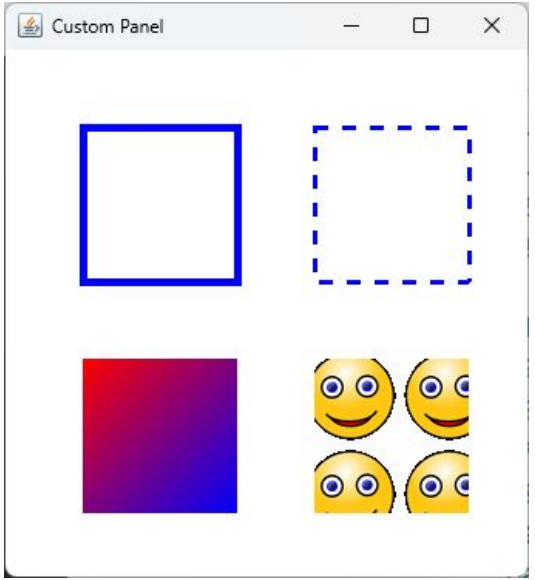
```
import java.awt.*;
import javax.swing.JPanel;
public class GraphicsPanel extends JPanel {
    public GraphicsPanel(int width, int height, Color backColor) {
        setPreferredSize(new Dimension(width, height));
        setBackground(backColor);
    }
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.red);
        g.fillOval(50, 50, 100, 100);
        g.setColor(new Color(128, 64, 128));
        g.fillOval(200, 50, 100, 100);
        g.setColor(new Color(0.2f, 0.5f, 0.2f, 0.5f));
        g.fillOval(250, 50, 100, 100);
    }
}
```

Color.black	0, 0, 0
Color.blue	0, 0, 255
Color.cyan	0, 255, 255
Color.orange	255, 200, 0
Color.white	255, 255, 255
Color.yellow	255, 255, 0



Line and Texture Styles

```
super.paintComponent(g);
Graphics2D g2d = (Graphics2D) g;
g2d.setColor(Color.blue);
g2d.setStroke(new BasicStroke(5));
g2d.drawRect(50, 50, 100, 100);
g2d.setStroke(new BasicStroke(3, BasicStroke.CAP_BUTT,
                BasicStroke.JOIN_BEVEL, 0, new float[] { 9 }, 0));
g2d.drawRect(200, 50, 100, 100);
g2d.setPaint(new GradientPaint(200, 50, Color.red, 300, 150, Color.blue));
g2d.fillRect(50, 200, 100, 100);
g2d.setPaint(new TexturePaint(image, new Rectangle.Double(0, 0, image.getWidth(),
image.getHeight())));
g2d.fillRect(200, 200, 100, 100);
```

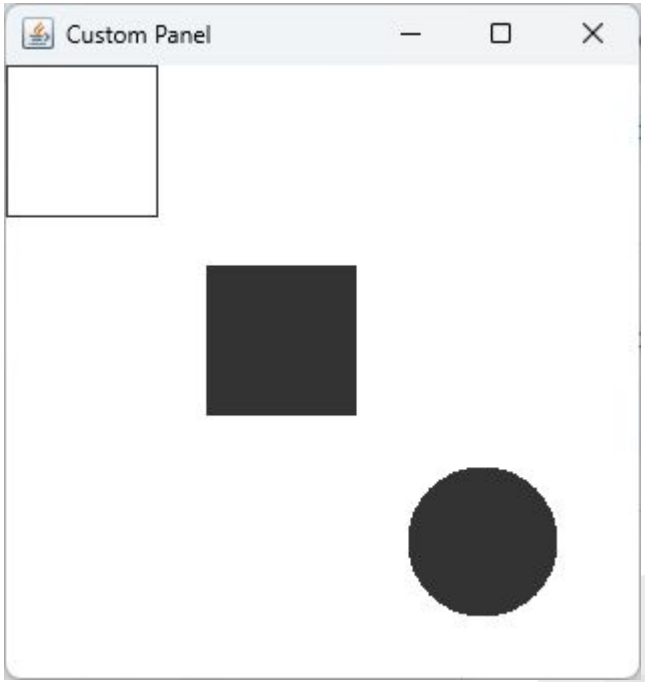


Fonts

```
super.paintComponent(g);
g.setFont(new Font("Arial", Font.BOLD | Font.ITALIC, 25));
g.setColor(Color.red);
g.drawString("Hello World", 50, 50);
```

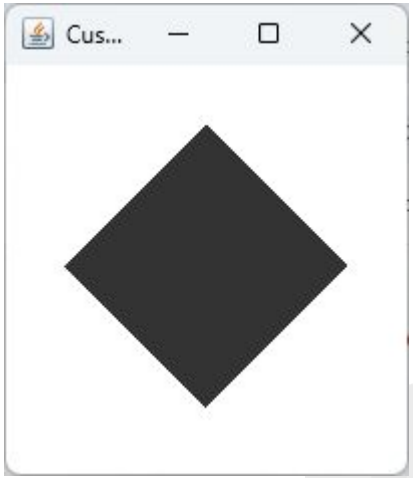
Translate

```
super.paintComponent(g);
Graphics2D g2d = (Graphics2D) g;
g2d.translate(100, 100);
g2d.fillRect(0, 0, 75, 75);
g2d.translate(100, 100);
g2d.fillOval(0, 0, 75, 75);
g2d.translate(-200, -200);
g2d.drawRect(0, 0, 75, 75);
```



Rotate

```
super.paintComponent(g);
Graphics2D g2d = (Graphics2D) g;
g2d.rotate(45 * Math.PI / 180.0, 100, 100);
g2d.fillRect(50, 50, 100, 100);
```



Animation

```
private int counter = 0;
public class AnimationProcessor extends TimerTask {
    public void run() {
        repaint();
    }
}
public GraphicsPanel(int width, int height, Color backColor) {
    setPreferredSize(new Dimension(width, height));
    setBackground(backColor);
    Timer timer = new Timer();
    timer.scheduleAtFixedRate(new AnimationProcessor(), (long) 1000, (long) 100);
}
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setFont(new Font("Arial", Font.BOLD, 20));
    g.setColor(Color.blue);
    g.drawString("Cunter : " + counter++, 50, 50);
}
```

Self Drawing Object

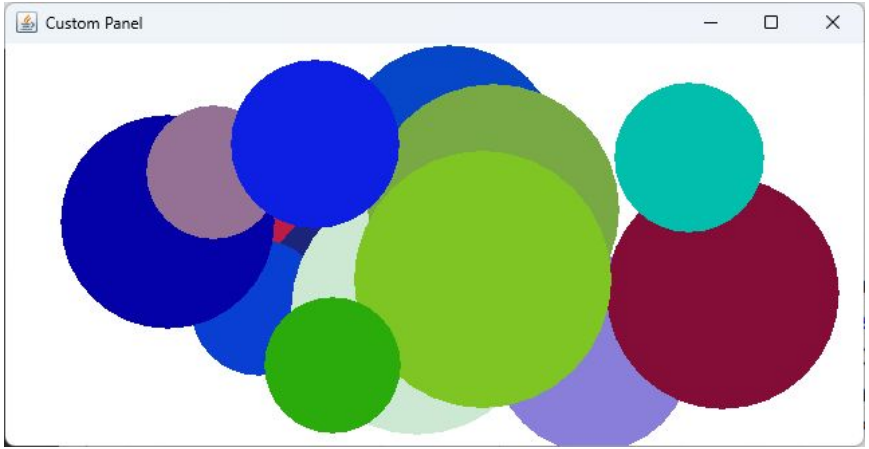
```
import java.awt.*;
class Circle {
    private int x, y, r;
    private Color c;
    public Circle(int x, int y, int r, Color c) {
        this.x = x; this.y = y;
        this.r = r; this.c = c;
    }
    public void Draw(Graphics g) {
        g.setColor(c);
        g.fillOval(x - r, y - r, 2 * r, 2 * r);
    }
    public void setPos(int x, int y) {
        this.x = x; this.y = y;
    }
}
```

Handling Mouse Events

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
public class GraphicsPanel extends JPanel implements MouseListener {
    private ArrayList<Circle> circles = new ArrayList<Circle>();
    private Random random = new Random();
    public GraphicsPanel(int width, int height, Color backColor) {
        setPreferredSize(new Dimension(width, height));
        setBackground(backColor);
        this.addMouseListener(this);
    }
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        for (Circle circle : circles)
            circle.Draw(g);
    }
    .......
}
```

Handling Mouse Events

```
public class GraphicsPanel extends JPanel implements MouseListener {
    ...
    public void mousePressed(MouseEvent e) {
        circles.add(new Circle(e.getX(), e.getY(), 50 + random.nextInt(50),
                new Color(random.nextInt(255), random.nextInt(255), random.nextInt(255))));
        paintComponent(getGraphics());
    }
    public void mouseClicked(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
}
```

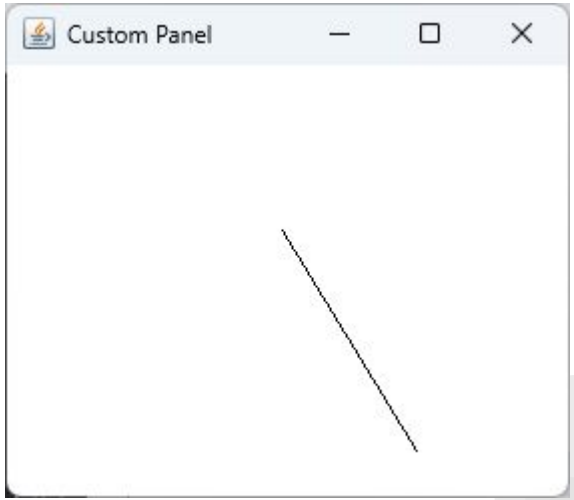


Handling Mouse Motion (Press and Move)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class GraphicsPanel extends JPanel implements MouseListener, MouseMotionListener {
    private Point point1 = null, point2 = null;
    public GraphicsPanel(int width, int height, Color backColor) {
        setPreferredSize(new Dimension(width, height));
        setBackground(backColor);
        addMouseListener(this);
        addMouseMotionListener(this);
    }
    public void paintComponent(Graphics page) {
        super.paintComponent(page);
        page.setColor(Color.black);
        if (point1 != null && point2 != null) {
            page.drawLine(point1.x, point1.y, point2.x, point2.y);
        }
    }
    ...
}
```

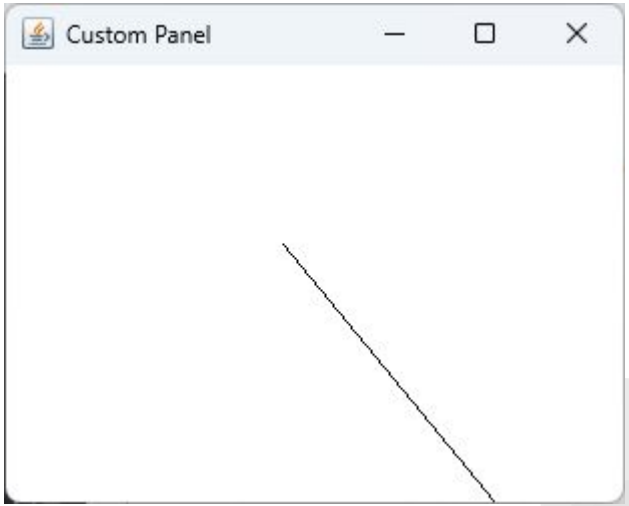
Handling Mouse Motion (Press and Move)

```
public class GraphicsPanel extends JPanel implements MouseListener, MouseMotionListener {
    ...
    public void mousePressed(MouseEvent event) {
        point1 = event.getPoint();
    }
    public void mouseClicked(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mouseDragged(MouseEvent e) {}
    public void mouseMoved(MouseEvent event) {
        point2 = event.getPoint();
        repaint();
    }
}
```



Handling Mouse Motion (Press and Drag)

```
public class GraphicsPanel extends JPanel implements MouseListener, MouseMotionListener {
    ...
    public void mousePressed(MouseEvent event) {
        point1 = event.getPoint();
    }
    public void mouseClicked(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mouseDragged(MouseEvent event) {
        point2 = event.getPoint();
        repaint();
    }
    public void mouseMoved(MouseEvent e) {}
}
```

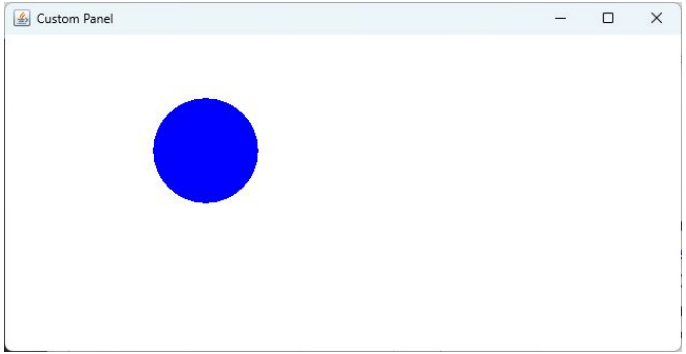


Handling Keyboard Events

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class GraphicsPanel extends JPanel implements KeyListener {
    private Circle circle = new Circle(0, 0, 50, Color.blue);
    public GraphicsPanel(int width, int height, Color backColor) {
        setPreferredSize(new Dimension(width, height));
        setBackground(backColor);
        this.addKeyListener(this);
        setFocusable(true);
    }
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        circle.Draw(g);
    }
    ...
}
```

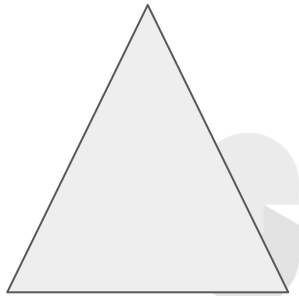
Handling Keyboard Events

```
public class GraphicsPanel extends JPanel implements KeyListener {
    ...
    public void keyPressed(KeyEvent e) {
        switch (e.getKeyCode()) {
            case KeyEvent.VK_UP: circle.moveBy(0, -10); break;
            case KeyEvent.VK_DOWN: circle.moveBy(0, 10); break;
            case KeyEvent.VK_LEFT: circle.moveBy(-10, 0); break;
            case KeyEvent.VK_RIGHT: circle.moveBy(10, 0); break;
        }
        paintComponent(getGraphics());
    }
    public void keyTyped(KeyEvent e) {}
    public void keyReleased(KeyEvent e) {}
}
```



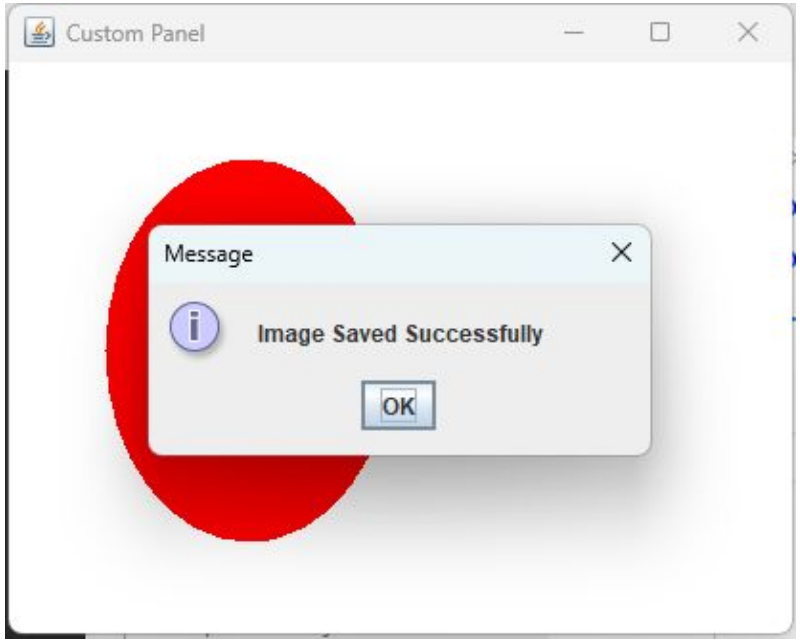
Saving Graphics to a File

```
import java.awt.*;
import java.awt.event.*;
import java.awt.image.BufferedImage;
import java.io.*;
import javax.imageio.ImageIO;
import javax.swing.*;
public class GraphicsPanel extends JPanel implements MouseListener {
    private BufferedImage image;
    public GraphicsPanel(int width, int height, Color backColor) {
        setPreferredSize(new Dimension(width, height));
        setBackground(backColor);
        this.addMouseListener(this);
    }
    protected void paintComponent(Graphics g) {
        image = new BufferedImage(this.getSize().width, this.getSize().height, BufferedImage.TYPE_INT_ARGB);
        Graphics ig = image.getGraphics();
        super.paintComponent(ig);
        ig.setColor(Color.red);
        ig.fillOval(50, 50, 150, 200);
        g.drawImage(image, 0, 0, this);
    }
    ...
}
```



Saving Graphics to a File

```
public class GraphicsPanel extends JPanel implements MouseListener {
    ...
    public void mouseClicked(MouseEvent e) {
        try {
            File outputfile = new File("d:\\saved.png");
            ImageIO.write(image, "png", outputfile);
            JOptionPane.showMessageDialog(this,
                    "Image Saved Successfully");
        } catch (IOException ex) {
        }
    }
    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
}
```



ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

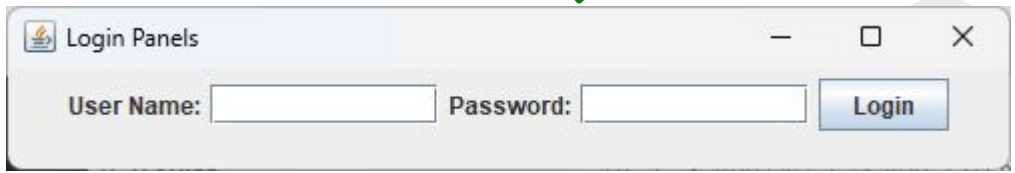
Java GUI

Trivial Input/Output GUI with Java

```
import javax.swing.*;
public class App {
    public static void main(String[] args) throws Exception {
        String numStr, result;
        int num, again;
        do {
            numStr = JOptionPane.showInputDialog("Enter an integer: ");
            num = Integer.parseInt(numStr);
            result = "That number is " + ((num % 2 == 0) ? "even" : "odd");
            JOptionPane.showMessageDialog(null, result);
            again = JOptionPane.showConfirmDialog(null, "Do Another?");
        } while (again == JOptionPane.YES_OPTION);
    }
}
```

Basic GUI: Username and Password Example

```
JFrame frame = new JFrame("Login Panels");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

//Panel Preparation Section
JPanel panel = new JPanel();
panel.setPreferredSize(new Dimension(500, 50));
panel.add(new JLabel("User Name:"));
panel.add(new JTextField(10));
panel.add(new JLabel("Password:"));
panel.add(new JTextField(10));
panel.add(new JButton("Login"));
frame.getContentPane().add(panel);

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



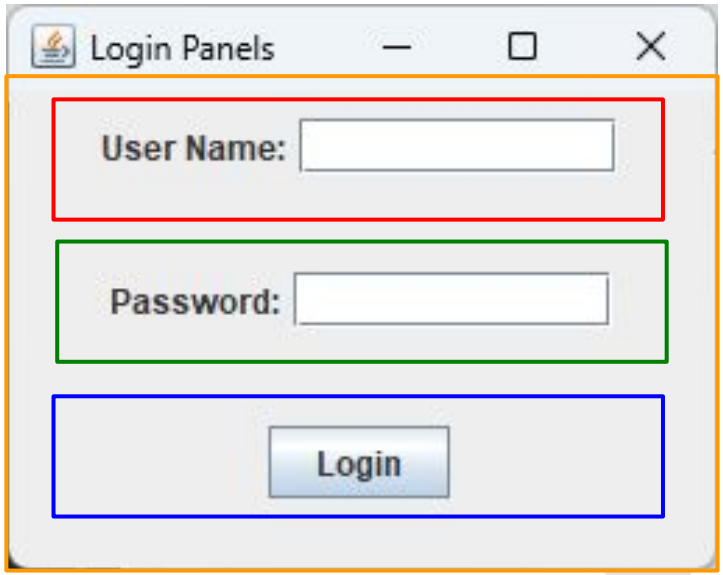
Nested Panels

```
//Name Panel
JPanel namePanel = new JPanel();
namePanel.setPreferredSize (new Dimension(200, 50));
namePanel.add (new JLabel ("User Name:"));
namePanel.add (new JTextField(10));
```

```
//Password
JPanel passwordPanel = new JPanel();
passwordPanel.setPreferredSize (new Dimension(200, 50));
passwordPanel.add (new JLabel ("Password:"));
passwordPanel.add (new JPasswordField(10));
```

```
//Action
JPanel actionPanel = new JPanel();
actionPanel.setPreferredSize (new Dimension(200, 50));
actionPanel.add (new JButton("Login"));
```

```
//Main Panel
JPanel mainPanel = new JPanel();
mainPanel.add(namePanel);
mainPanel.add(passwordPanel);
mainPanel.add(actionPanel);
frame.getContentPane().add(mainPanel);
```



Panels Layouts

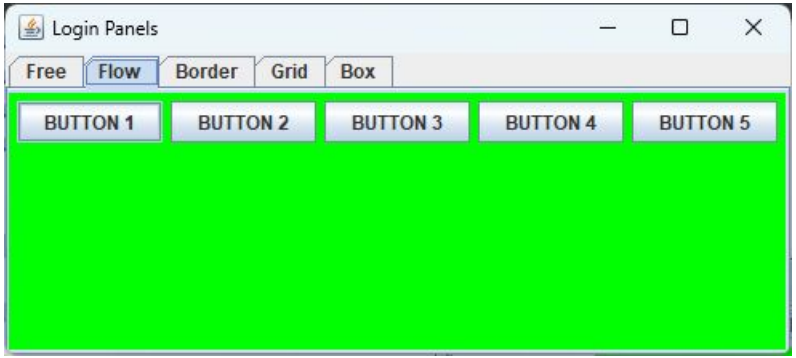
```
JTabbedPane tappedPanel = new JTabbedPane();
tappedPanel .addTab("Free", new FreePanel());
tappedPanel .addTab("Flow", new FlowPanel());
tappedPanel .addTab("Border", new BorderPanel());
tappedPanel .addTab("Grid", new GridPanel());
tappedPanel .addTab("Box", new BoxPanel());
frame.getContentPane().add(tappedPanel);
```

Free Layout

```
import java.awt.*;
import javax.swing.*;
public class FreePanel extends JPanel {
    public FreePanel() {
        setLayout(null);
        setBackground(Color.green);
        JButton b1 = new JButton("BUTTON 1");
        JButton b2 = new JButton("BUTTON 2");
        JButton b3 = new JButton("BUTTON 3");
        add(b1); add(b2); add(b3);
        b1.setBounds(10, 10, 90, 40);
        b2.setBounds(110, 60, 150, 40);
        b3.setBounds(200, 10, 100, 40);
    }
}
```

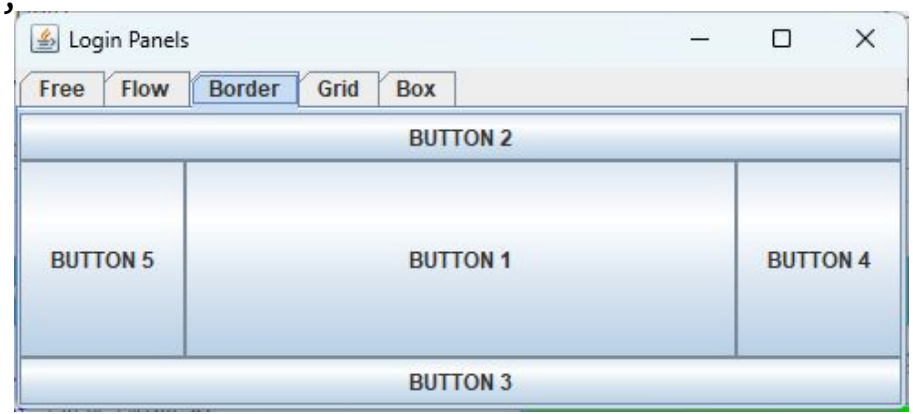
Flow Layout

```
import java.awt.*;
import javax.swing.*;
public class FlowPanel extends JPanel {
    public FlowPanel() {
        setLayout(new FlowLayout());
        setBackground(Color.green);
        JButton b1 = new JButton("BUTTON 1");
        JButton b2 = new JButton("BUTTON 2");
        JButton b3 = new JButton("BUTTON 3");
        JButton b4 = new JButton("BUTTON 4");
        JButton b5 = new JButton("BUTTON 5");
        add(b1); add(b2); add(b3); add(b4); add(b5);
    }
}
```



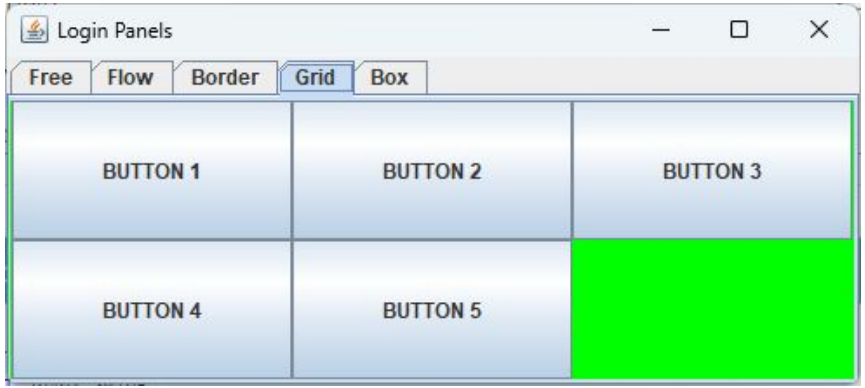
Border Layout

```
import java.awt.*;
import javax.swing.*;
public class BorderPanel extends JPanel {
    public BorderPanel() {
        setLayout(new BorderLayout());
        setBackground(Color.green);
        JButton b1 = new JButton("BUTTON 1");
        JButton b2 = new JButton("BUTTON 2");
        JButton b3 = new JButton("BUTTON 3");
        JButton b4 = new JButton("BUTTON 4");
        JButton b5 = new JButton("BUTTON 5");
        add(b1, BorderLayout.CENTER);
        add(b2, BorderLayout.NORTH);
        add(b3, BorderLayout.SOUTH);
        add(b4, BorderLayout.EAST);
        add(b5, BorderLayout.WEST);
    }
}
```



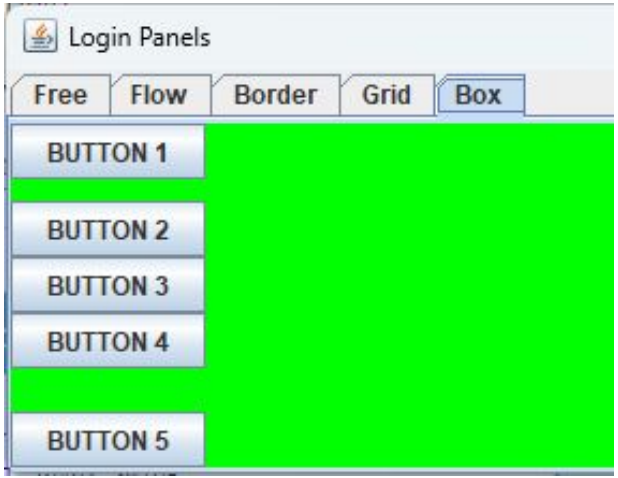
Grid Layout

```
import java.awt.*;
import javax.swing.*;
public class GridPanel extends JPanel {
    public GridPanel() {
        setLayout(new GridLayout(2, 3));
        setBackground(Color.green);
        JButton b1 = new JButton("BUTTON 1");
        JButton b2 = new JButton("BUTTON 2");
        JButton b3 = new JButton("BUTTON 3");
        JButton b4 = new JButton("BUTTON 4");
        JButton b5 = new JButton("BUTTON 5");
        add(b1); add(b2); add(b3); add(b4); add(b5);
    }
}
```



Box Layout

```
import java.awt.*;
import javax.swing.*;
public class BoxPanel extends JPanel {
    public BoxPanel() {
        setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));
        setBackground(Color.green);
        JButton b1 = new JButton("BUTTON 1");
        JButton b2 = new JButton("BUTTON 2");
        JButton b3 = new JButton("BUTTON 3");
        JButton b4 = new JButton("BUTTON 4");
        JButton b5 = new JButton("BUTTON 5");
        add(b1);
        add(Box.createRigidArea(new Dimension(0, 10)));
        add(b2);
        add(Box.createVerticalGlue());
        add(b3);
        add(b4);
        add(Box.createRigidArea(new Dimension(0, 20)));
        add(b5);
    }
}
```

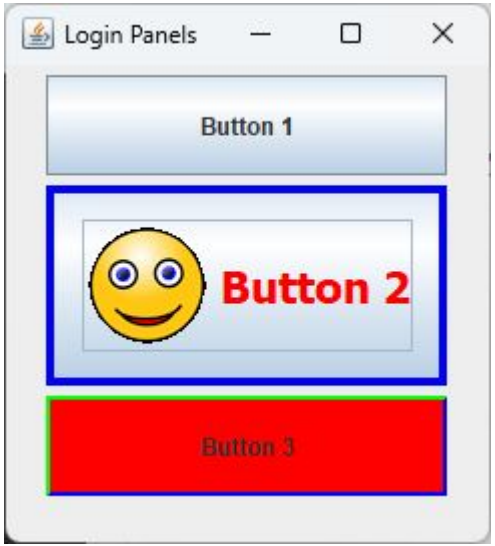


Panel Borders

```
JPanel panel = new JPanel();
panel.setLayout(new GridLayout(0, 2, 5, 10));
panel.setBorder(BorderFactory.createEmptyBorder(8, 8, 8, 8));

JPanel p1 = new JPanel();
p1.setBorder(BorderFactory.createLineBorder(Color.red, 3));
p1.add(new JLabel("Line Border"));
panel.add(p1);

JPanel p2 = new JPanel();
p2.setBorder(BorderFactory.createEtchedBorder());
p2.add(new JLabel("Etched Border"));
panel.add(p2);

JPanel p3 = new JPanel();
p3.setBorder(BorderFactory.createRaisedBevelBorder());
p3.add(new JLabel("Raised Bevel Border"));
panel.add(p3);

JPanel p4 = new JPanel();
p4.setBorder(BorderFactory.createTitledBorder("Panel Title"));
p4.add(new JLabel("Title Border"));
panel.add(p4);

frame.getContentPane().add(panel);
```

Changing Controls Shape

```
JButton button1 = new JButton("Button 1");
button1.setPreferredSize(new Dimension(200, 50));
panel.add(button1);

JButton button2 = new JButton("Button 2");
button2.setPreferredSize(new Dimension(200, 100));
button2.setFont(new Font("Tahoma", Font.BOLD, 22));
button2.setForeground(Color.red);
button2.setBorder(BorderFactory.createLineBorder(Color.blue, 4));
button2.setIcon(new ImageIcon("smile.gif"));
panel.add(button2);

JButton button3 = new JButton("Button 3");
button3.setPreferredSize(new Dimension(200, 50));
button3.setBackground(Color.red);
button3.setBorder(BorderFactory.createBevelBorder(BevelBorder.LOWERED, Color.blue, Color.green));
panel.add(button3);
```

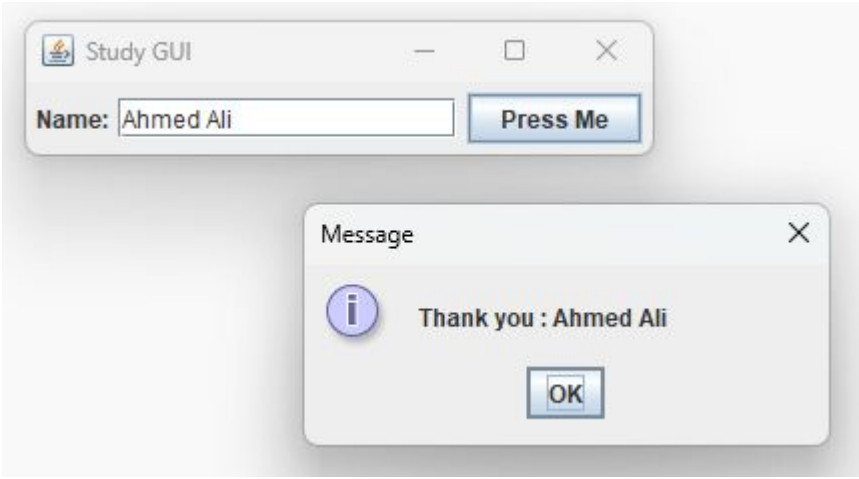


Handling Events Example

```
public class GUIPanel extends JPanel implements ActionListener {
    private JTextField textField;
    private JButton button;
    public GUIPanel() {
        add(new JLabel("Name:"));
        add(textField = new JTextField("", 15));
        add(button = new JButton("Press Me"));
        button.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e) {
        if (textField.getText().trim().isEmpty()) {
            JOptionPane.showMessageDialog(this, "You must enter a value");
        } else {
            JOptionPane.showMessageDialog(this, "Thank you : " + textField.getText());
            System.exit(0);
        }
    }
}
```

Handling Events Example

```
import javax.swing.*;
public class App {
    public static void main(String[] args) throws Exception {
        JFrame frame = new JFrame("Study GUI");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        GUIPanel panel = new GUIPanel();
        frame.getContentPane().add(panel);
        frame.pack();
        frame.setVisible(true);
    }
}
```



```
public class GUIPanel extends JPanel implements ActionListener {
    private JTextField xTextField, yTextField, rTextField;
    private JButton addButton, subButton;
    public GUIPanel() {
        add(new JLabel("Name:"));
        add(xTextField = new JTextField("0.0", 6));
        add(yTextField = new JTextField("0.0", 6));
        add(addButton = new JButton("Add"));
        addButton.addActionListener(this);
        add(subButton = new JButton("Sub"));
        subButton.addActionListener(this);
        add(rTextField = new JTextField("0.0", 6));
    }
    public void actionPerformed(ActionEvent e) {
        double x = Double.parseDouble(xTextField.getText());
        double y = Double.parseDouble(yTextField.getText());
        double r = 0;
        if (e.getSource() == addButton) r = x + y;
        else r = x - y;
        rTextField.setText("" + r);
    }
}
```

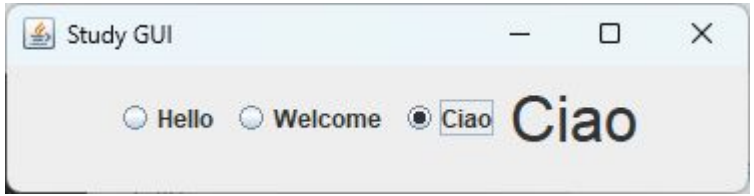
Example: Simple Calculator

Example:
Working with Checkboxes

```
te JLabel title;
private JCheckBox bold, italic;
public class CheckBoxListener implements ItemListener {
    public void itemStateChanged(ItemEvent e) {
        int style = 0;
        if (bold.isSelected()) style |= Font.BOLD;
        if (italic.isSelected()) style |= Font.ITALIC;
        title.setFont(new Font("Helvetica", style, 32));
    }
}
public GUIPanel() {
    title = new JLabel("Welcome to Java World");
    title.setFont(new Font("Helvetica", Font.PLAIN, 32)); add(title);
    bold = new JCheckBox("Bold"); add(bold);
    italic = new JCheckBox("Italic"); add(italic);
    CheckBoxListener listener = new CheckBoxListener();
    bold.addItemListener(listener);
    italic.addItemListener(listener);
    setPreferredSize(new Dimension(370, 100));
}
```

Example:
Working with Radio Buttons

```
private JLabel title;
private JRadioButton radio1, radio2, radio3;
public class RadioListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        title.setText(((JRadioButton) e.getSource()).getText());
    }
}
public GUIPanel() {
    ButtonGroup buttonGroup = new ButtonGroup();
    add(radio1 = new JRadioButton("Hello"));
    add(radio2 = new JRadioButton("Welcome"));
    add(radio3 = new JRadioButton("Ciao"));
    buttonGroup.add(radio1);
    buttonGroup.add(radio2);
    buttonGroup.add(radio3);
    title = new JLabel("");
    title.setFont(new Font("Helvetica", Font.PLAIN, 32));
    add(title);
    RadioListener listener = new RadioListener();
    radio1.addActionListener(listener);
    radio2.addActionListener(listener);
    radio3.addActionListener(listener);
    setPreferredSize(new Dimension(370, 100));
}
```



Example:
Working with Slider

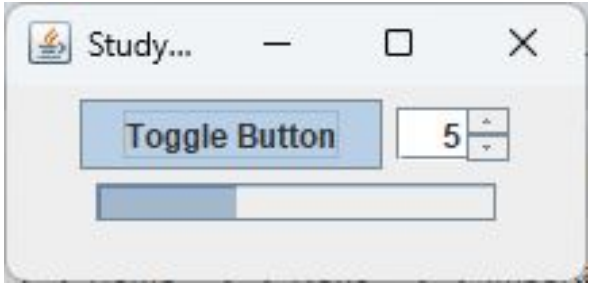
```
private JLabel label;
private JSlider slider;
public GUIPanel() {
    add(label = new JLabel("Welcome to Java"));
    add(slider = new JSlider(6, 40));
    slider.addChangeListener(new ChangeListener() {
        public void stateChanged(ChangeEvent e) {
            label.setFont(new Font("Arial", 0, slider.getValue()));
        }
    });
    setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));
    setPreferredSize(new Dimension(370, 100));
}
```

More Controls

```
public GUIPanel() {
    setPreferredSize(new Dimension(300, 200));
    add(new JLabel("Label:"));
    add(new JTextField("Text Field", 15));
    add(new JPasswordField(10));
    add(new JButton("Button"));
    String values[] = { "RED", "GREEN", "YELLOW", "GREEN", "CYAN", "ORANGE" };
    JComboBox comboBox = new JComboBox(values);
    comboBox.setSelectedItem("YELLOW");
    add(comboBox);
    JRadioButton radio1 = new JRadioButton("Male", true);
    JRadioButton radio2 = new JRadioButton("Female", false);
    ButtonGroup genderGroup = new ButtonGroup();
    genderGroup.add(radio1);
    genderGroup.add(radio2);
    JPanel genderPanel = new JPanel();
    genderPanel.setLayout(new GridLayout(2, 1));
    genderPanel.add(radio1);
    genderPanel.add(radio2);
    add(genderPanel);
    JCheckBox checkBox = new JCheckBox("Chech Box");
    checkBox.setSelected(true);
    add(checkBox);
}
```

More Controls

```
public GUIPanel() {
    setPreferredSize(new Dimension(300, 200));
    JToggleButton toggleButton = new JToggleButton("Toggle Button");
    add(toggleButton);
    toggleButton.setSelected(true);
    JSpinner spinner = new JSpinner(new SpinnerNumberModel(0, 0, 10, 1));
    add(spinner);
    spinner.setValue(5);
    JProgressBar progressBar = new JProgressBar(0, 100);
    add(progressBar);
    progressBar.setValue(35);
    progressBar.setToolTipText("Operation Progress...");
}
```

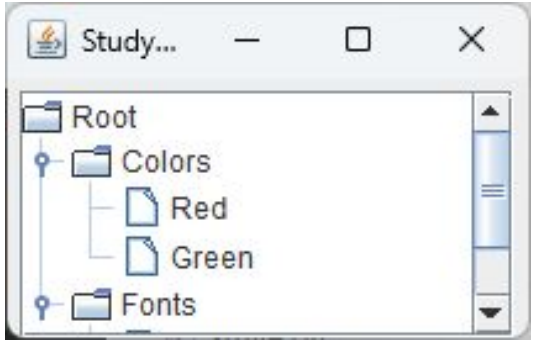


Working with Tables

```
public class GUIPanel extends JPanel {
    public GUIPanel() {
        JTable table = new JTable(5, 5);
        table.setValueAt("Hello", 2, 3);
        add(table);
    }
}
```

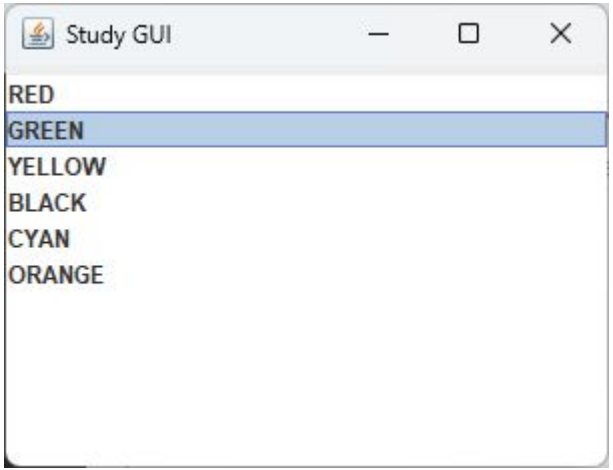
Working with Tables and Scroll

```
setPreferredSize(new Dimension(200, 100));
DefaultMutableTreeNode root = new DefaultMutableTreeNode("Root");
DefaultMutableTreeNode colors = new DefaultMutableTreeNode("Colors");
colors.add(new DefaultMutableTreeNode("Red"));
colors.add(new DefaultMutableTreeNode("Green"));
DefaultMutableTreeNode fonts = new DefaultMutableTreeNode("Fonts");
fonts.add(new DefaultMutableTreeNode("Arial"));
fonts.add(new DefaultMutableTreeNode("Tahoma"));
root.add(colors);
root.add(fonts);
JTree tree = new JTree(root);
JScrollPane scrollingTree = new JScrollPane(tree);
scrollingTree.setPreferredSize(new Dimension(200,100));
add(scrollingTree);
```



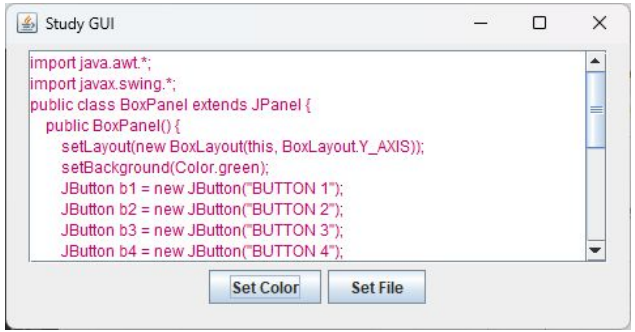
Working with Lists

```
setPreferredSize(new Dimension(300, 200));
DefaultListModel listModel = new DefaultListModel();
listModel.addElement("RED");
listModel.addElement("GREEN");
listModel.addElement("YELLOW");
listModel.addElement("BLACK");
listModel.addElement("CYAN");
listModel.addElement("ORANGE");
JList list = new JList(listModel);
list.setSelectedValue("GREEN", true);
list.setPreferredSize(new Dimension(300, 200));
add(list);
```



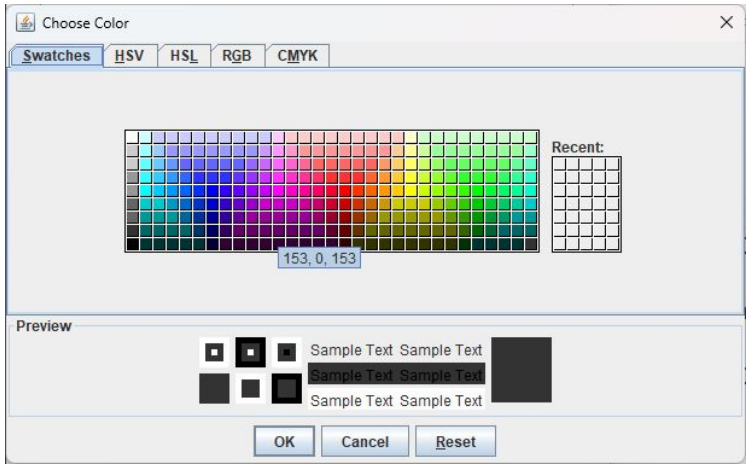
File and Color Chooser

```
public class GUIPanel extends JPanel {
    private JTextArea textArea;
    private JButton colorButton, fileButton;
    public GUIPanel() {
        textArea = new JTextArea("Welcome to Java", 10, 40);
        textArea.setLineWrap(true);
        JScrollPane scrollPane = new JScrollPane(textArea);
        add(scrollPane);
        add(colorButton = new JButton("Set Color"));
        add(fileButton = new JButton("Set File"));
        colorButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) { ChooseColor(); }
        });
        fileButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) { ChooseFile(); }
        });
        setPreferredSize(new Dimension(500, 400));
    }
    . . .
}
```



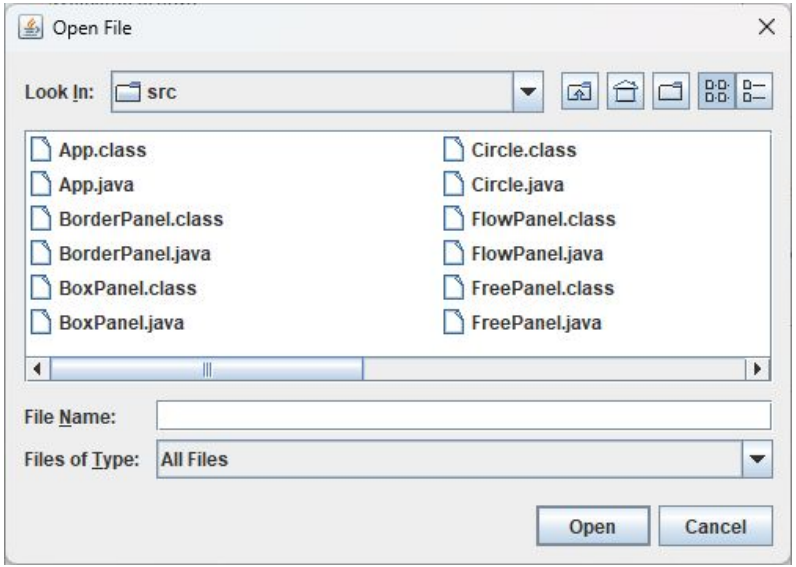
File and Color Chooser

```
public class GUIPanel extends JPanel {
    ...
    public void ChooseColor() {
        Color color = JColorChooser.showDialog(this, "Choose Color", textArea.getForeground());
        textArea.setForeground(color);
    }
    ...
}
```



File and Color Chooser

```
public class GUIPanel extends JPanel {
    ...
    public void ChooseFile() {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setDialogTitle("Open File");
        fileChooser.setFileSelectionMode(JFileChooser.FILES_ONLY);
        fileChooser.setCurrentDirectory(new File("."));
        int result = fileChooser.showOpenDialog(this);
        if (result == JFileChooser.APPROVE_OPTION) {
            File file = fileChooser.getSelectedFile();
            try {
                Scanner scanner = new Scanner(file);
                String text = "";
                while (scanner.hasNext())
                    text += scanner.nextLine() + "\n";
                textArea.setText(text);
            } catch (Exception ex) {
            }
        }
    }
}
```

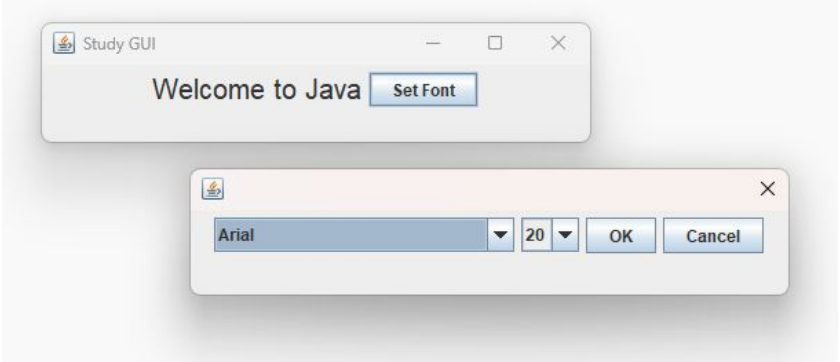


Change Font Example

```
public class FontChooserPanel extends JPanel implements ActionListener {
    private JButton okButton, cancelButton;
    private Font selectedFont;
    private JDialog parent;
    private JComboBox fontsBox, sizesBox;
    public FontChooserPanel(Font font) {
        this.parent = parent;
        selectedFont = font;
        String[] fonts = GraphicsEnvironment.getLocalGraphicsEnvironment().getAvailableFontFamilyNames();
        String[] sizes = { "8", "10", "12", "14", "16", "20", "24", "32" };
        add(fontsBox = new JComboBox(fonts));
        fontsBox.setSelectedItem(font.getFontName());
        add(sizesBox = new JComboBox(sizes));
        sizesBox.setSelectedItem("" + font.getSize());
        add(okButton = new JButton("OK"));
        add(cancelButton = new JButton("Cancel"));
        okButton.addActionListener(this);
        cancelButton.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == okButton)
            selectedFont = new Font((String) fontsBox.getSelectedItem(), 0,
                    Integer.parseInt((String) sizesBox.getSelectedItem()));
        getRootPane().getParent().setVisible(false);
    }
    public Font getSelectedFont() { return selectedFont;}
}
```

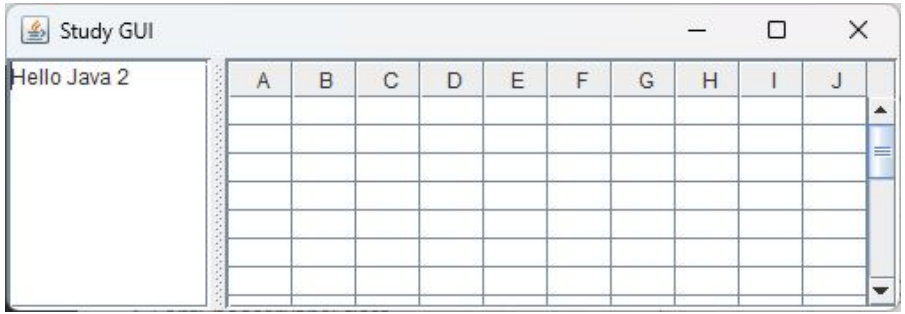
Change Font Example

```
public class GUIPanel extends JPanel implements ActionListener {
    private JLabel label;
    private JButton fontButton;
    public GUIPanel() {
        add(label = new JLabel("Welcome to Java"));
        add(fontButton = new JButton("Set Font"));
        fontButton.addActionListener(this);
        setPreferredSize(new Dimension(370, 100));
        label.setFont(new Font("Arial", 0, 10));
    }
    public void actionPerformed(ActionEvent e) {
        FontChooserPanel fontChooserPanel = new FontChooserPanel(label.getFont());
        JDialog dialog = new JDialog();
        dialog.setLocation(new Point(100, 100));
        dialog.setSize(new Dimension(450, 100));
        dialog.getContentPane().add(fontChooserPanel);
        dialog.setModal(true);
        dialog.setVisible(true);
        label.setFont(fontChooserPanel.getSelectedFont());
    }
}
```



Splitter Panel

```
public class GUIPanel extends JPanel {
    public GUIPanel() {
        JTextArea textArea = new JTextArea("Hello Java 2");
        JScrollPane rightScrollPane = new JScrollPane(textArea);
        JPanel rightPanel = new JPanel(new GridLayout(1, 1));
        rightPanel.add(rightScrollPane);
        JTable table = new JTable(20, 10);
        JScrollPane leftScrollPane = new JScrollPane(table);
        JPanel leftPanel = new JPanel(new GridLayout(1, 1));
        leftPanel.add(leftScrollPane);
        JSplitPane splitPane = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT);
        splitPane.add(rightPanel);
        splitPane.add(leftPanel);
        setPreferredSize(new Dimension(300, 200));
        setLayout(new GridLayout(1, 1));
        add(splitPane);
    }
}
```

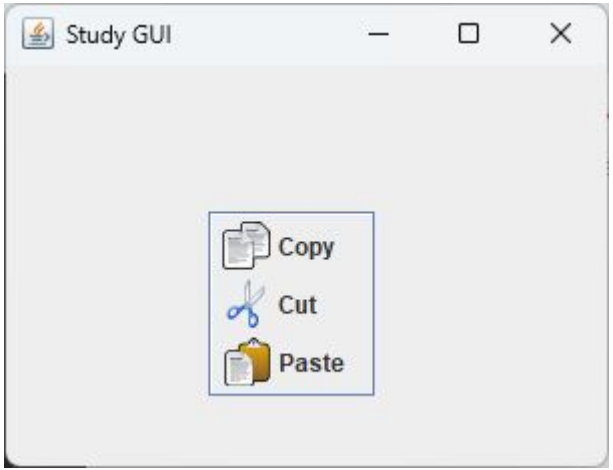


Using Menu

```
public class GUIPanel extends JPanel implements ActionListener {
    private JMenuBar menuBar;
    public GUIPanel(JFrame frame) {
        setPreferredSize(new Dimension(370, 100));
        menuBar = new JMenuBar();
        JMenu menu;
        JMenuItem menuItem;
        ButtonGroup group = new ButtonGroup();
        menuBar.add(menu = new JMenu("Main"));
        menu.add(menuItem = new JMenuItem("Open"));
        menuItem.addActionListener(this);
        menu.add(menuItem = new JMenuItem("Close"));
        menuItem.addActionListener(this);
        menuBar.add(menu = new JMenu("Options"));
        menu.add(menuItem = new JCheckBoxMenuItem("Check 1"));
        menuItem.addActionListener(this);
        menu.add(menuItem = new JCheckBoxMenuItem("Check 2"));
        menuItem.addActionListener(this);
        menu.add(new JSeparator());
        menu.add(menuItem = new JRadioButtonMenuItem("Option A"));
        menuItem.addActionListener(this);
        group.add(menuItem);
        menu.add(menuItem = new JRadioButtonMenuItem("Option B"));
        menuItem.addActionListener(this);
        group.add(menuItem);
        frame.getRootPane().setJMenuBar(menuBar);
    }
    public void actionPerformed(ActionEvent e) {
        JMenuItem menuItem = (JMenuItem) e.getSource();
        JOptionPane.showMessageDialog(this, menuItem.getText());
    }
}
```

Popup Menu

```
public class GUIPanel extends JPanel {
    JPopupMenu popupMenu;
    public GUIPanel() {
        setPreferredSize(new Dimension(300, 200));
        setLayout(new GridLayout(1, 1));
        popupMenu = new JPopupMenu();
        popupMenu.add(new JMenuItem("Copy", new ImageIcon("copy.png")));
        popupMenu.add(new JMenuItem("Cut", new ImageIcon("cut.png")));
        popupMenu.add(new JMenuItem("Paste", new ImageIcon("paste.png")));
        addMouseListener(new MYMouseAdapter());
    }
    class MYMouseAdapter extends MouseAdapter {
        public void mousePressed(MouseEvent e) { checkPopup(e); }
        public void mouseClicked(MouseEvent e) { checkPopup(e); }
        public void mouseReleased(MouseEvent e) { checkPopup(e); }
        private void checkPopup(MouseEvent e) {
            if (e.isPopupTrigger()) {
                popupMenu.show(GUIPanel.this, e.getX(), e.getY());
            }
        }
    }
}
```



Toolbar

```
public class GUIPanel extends JPanel implements ActionListener {
    private JToolBar toolBar;
    JButton button1, button2;
    JCheckBox cBox1, cBox2;
    public GUIPanel() {
        setPreferredSize(new Dimension(370, 100));
        toolBar = new JToolBar();
        toolBar.add(button1 = new JButton(new ImageIcon("open.png")));
        button1.addActionListener(this);
        toolBar.add(button2 = new JButton(new ImageIcon("close.png")));
        button2.addActionListener(this);
        toolBar.addSeparator();
        toolBar.add(cBox1 = new JCheckBox("Option 1"));
        cBox1.addActionListener(this);
        toolBar.add(cBox2 = new JCheckBox("Option 2"));
        cBox2.addActionListener(this);
        setLayout(new BorderLayout());
        add(toolBar, BorderLayout.NORTH);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == button1) JOptionPane.showMessageDialog(this, "Open Pressed");
        if (e.getSource() == button2) JOptionPane.showMessageDialog(this, "Close Pressed");
        if (e.getSource() == cBox1) JOptionPane.showMessageDialog(this, "Option 1 Pressed");
        if (e.getSource() == cBox2) JOptionPane.showMessageDialog(this, "Option 2 Pressed");
    }
}
```