

[Print Window](#)   [Close Window](#)

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

public class Assignment06 implements ActionListener
{
    private JFrame frame;
    private JLabel instructionLabel;
    private JPanel leftPanel;
    private JPanel rightPanel;
    private JButton wrapButton, doNotWrapButton;
    private JButton emptyButton, clearButton;
    private JTextArea textArea;
    private JScrollPane scroller;
    private JButton scrollButton, scrollVerticalButton,
        scrollHorizontalButton, doNotScrollButton;

    public static void main (String[] args)
    {
        Assignment06 guiWindow = new Assignment06();
        guiWindow.setUp();
    }

    public void setUp()
    {
        frame = new JFrame("Assignment 6");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPane = frame.getContentPane();
        contentPane.setLayout(new BorderLayout());

        instructionLabel = new JLabel("Use the buttons to " +
            "control the layout of the text you type.");
        contentPane.add(instructionLabel,BorderLayout.NORTH);

        leftPanel = new JPanel();
        leftPanel.setLayout(new GridLayout(4,1));
        wrapButton = new JButton("Wrap Text");
        emptyButton = new JButton();
        clearButton = new JButton("Clear Text");
        doNotWrapButton = new JButton("Do Not Wrap Text");
        wrapButton.addActionListener(this);
        doNotWrapButton.addActionListener(this);
        clearButton.addActionListener(this);
        leftPanel.add(wrapButton);
        leftPanel.add(doNotWrapButton);
        leftPanel.add(emptyButton);
        leftPanel.add(clearButton);
        contentPane.add(leftPanel,BorderLayout.WEST);

        textArea = new JTextArea(5,15);
        scroller = new JScrollPane(textArea);
        scroller.setVerticalScrollBarPolicy(
            ScrollPaneConstants.VERTICAL_SCROLLBAR_NEVER);
        scroller.setHorizontalScrollBarPolicy(
            ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
        contentPane.add(scroller,BorderLayout.CENTER);

        rightPanel = new JPanel();
        rightPanel.setLayout(new GridLayout(4,1));
        scrollButton = new JButton("Scroll Both Ways");
        scrollVerticalButton = new JButton("Scroll Vertically");
```

```

        scrollHorizontalButton = new JButton(
            "Scroll Horizontally");
        doNotScrollButton = new JButton("Do Not Scroll");
        scrollButton.addActionListener(this);
        scrollVerticalButton.addActionListener(this);
        scrollHorizontalButton.addActionListener(this);
        doNotScrollButton.addActionListener(this);
        rightPanel.add(scrollVerticalButton);
        rightPanel.add(scrollHorizontalButton);
        rightPanel.add(scrollButton);
        rightPanel.add(doNotScrollButton);
        contentPane.add(rightPanel, BorderLayout.EAST);

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

    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource() == wrapButton)
            wrapText();
        else if (e.getSource() == doNotWrapButton)
            doNotWrapText();
        else if (e.getSource() == clearButton)
            clearText();
        else if (e.getSource() == scrollButton)
            scrollBothWays();
        else if (e.getSource() == scrollVerticalButton)
            scrollVertical();
        else if (e.getSource() == scrollHorizontalButton)
            scrollHorizontal();
        else if (e.getSource() == doNotScrollButton)
            doNotScroll();
        else
            textArea.append("Should not get here!\n");
    }

    private void wrapText()
    {
        textArea.setLineWrap(true);
    }

    private void doNotWrapText()
    {
        textArea.setLineWrap(false);
    }

    private void clearText()
    {
        textArea.setText("");
    }

    private void scrollBothWays()
    {
        scroller.setVerticalScrollBarPolicy(
            ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
        scroller.setHorizontalScrollBarPolicy(
            ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);
    }

    private void scrollVertical()
    {
        scroller.setVerticalScrollBarPolicy(
            ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
        scroller.setHorizontalScrollBarPolicy(

```

```
        ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
    }

    private void scrollHorizontal()
    {
        scroller.setVerticalScrollBarPolicy(
            ScrollPaneConstants.VERTICAL_SCROLLBAR_NEVER);
        scroller.setHorizontalScrollBarPolicy(
            ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);
    }

    private void doNotScroll()
    {
        scroller.setVerticalScrollBarPolicy(
            ScrollPaneConstants.VERTICAL_SCROLLBAR_NEVER);
        scroller.setHorizontalScrollBarPolicy(
            ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
    }
}
```

[Print Window](#)   [Close Window](#)