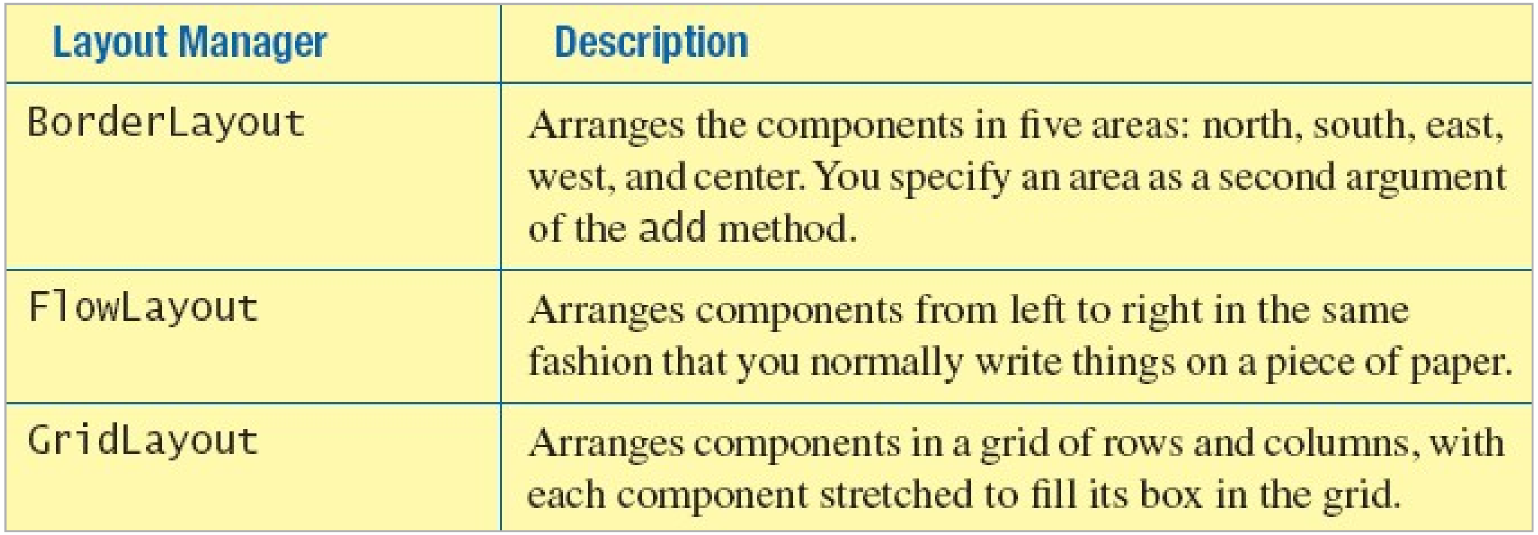
13.2B Basic Swing Details - Layout Managers

- Layout Manager—an object that decides how components will be arranged in a container
- Used because containers can change size
- Some types of layout managers:
  - » `BorderLayout`
  - » `FlowLayout`
  - » `GridLayout`
- Each type of layout manager has rules about how to rearrange components when the size or shape of the container changes.

Layout Managers

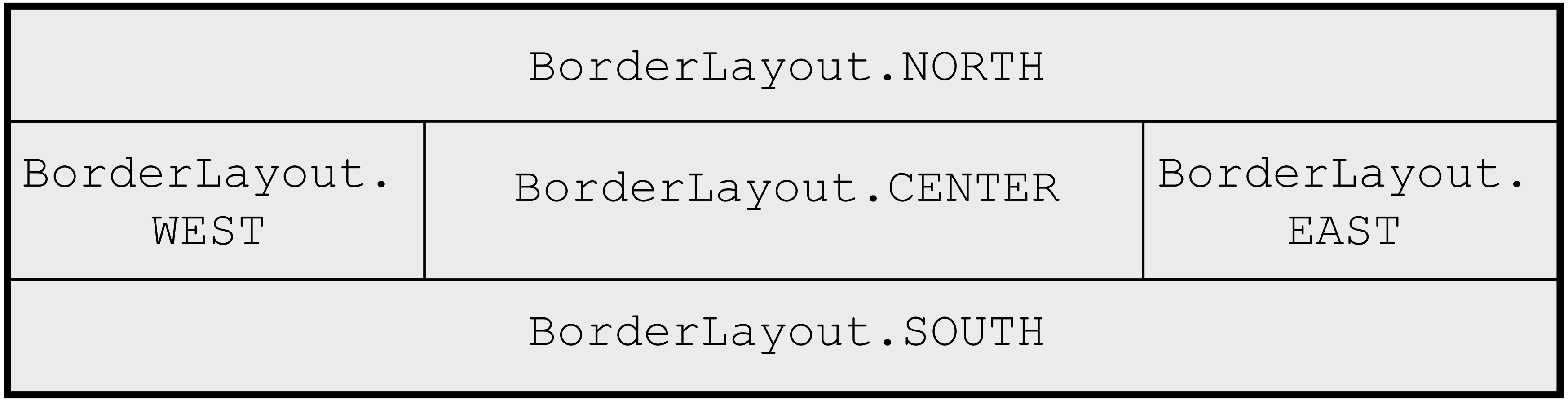
- Figure 13.7, some layout managers



Layout Manager	Description
`BorderLayout`	Arranges the components in five areas: north, south, east, west, and center. You specify an area as a second argument of the `add` method.
`FlowLayout`	Arranges components from left to right in the same fashion that you normally write things on a piece of paper.
`GridLayout`	Arranges components in a grid of rows and columns, with each component stretched to fill its box in the grid.

The Border Layout Manager

Five regions that can each have one component added to them:



BorderLayout.NORTH

BorderLayout.WEST

BorderLayout.CENTER

BorderLayout.EAST

BorderLayout.SOUTH

```
content.setLayout(new BorderLayout());
. . .
content.add(label1, BorderLayout.NORTH);
```

The CENTER region grows the most when the container grows and shrinks the most when the container shrinks

Listing 13.7 Using the BorderLayout Manager - BorderLayoutDemo.java

```
 //Listing 13.7 Using the BorderLayout Manager

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

/**
 Simple demonstration of using a layout manager
 to arrange labels.
*/
public class BorderLayoutDemo extends JFrame
{
    public static final int WIDTH = 300;
    public static final int HEIGHT = 200;
```

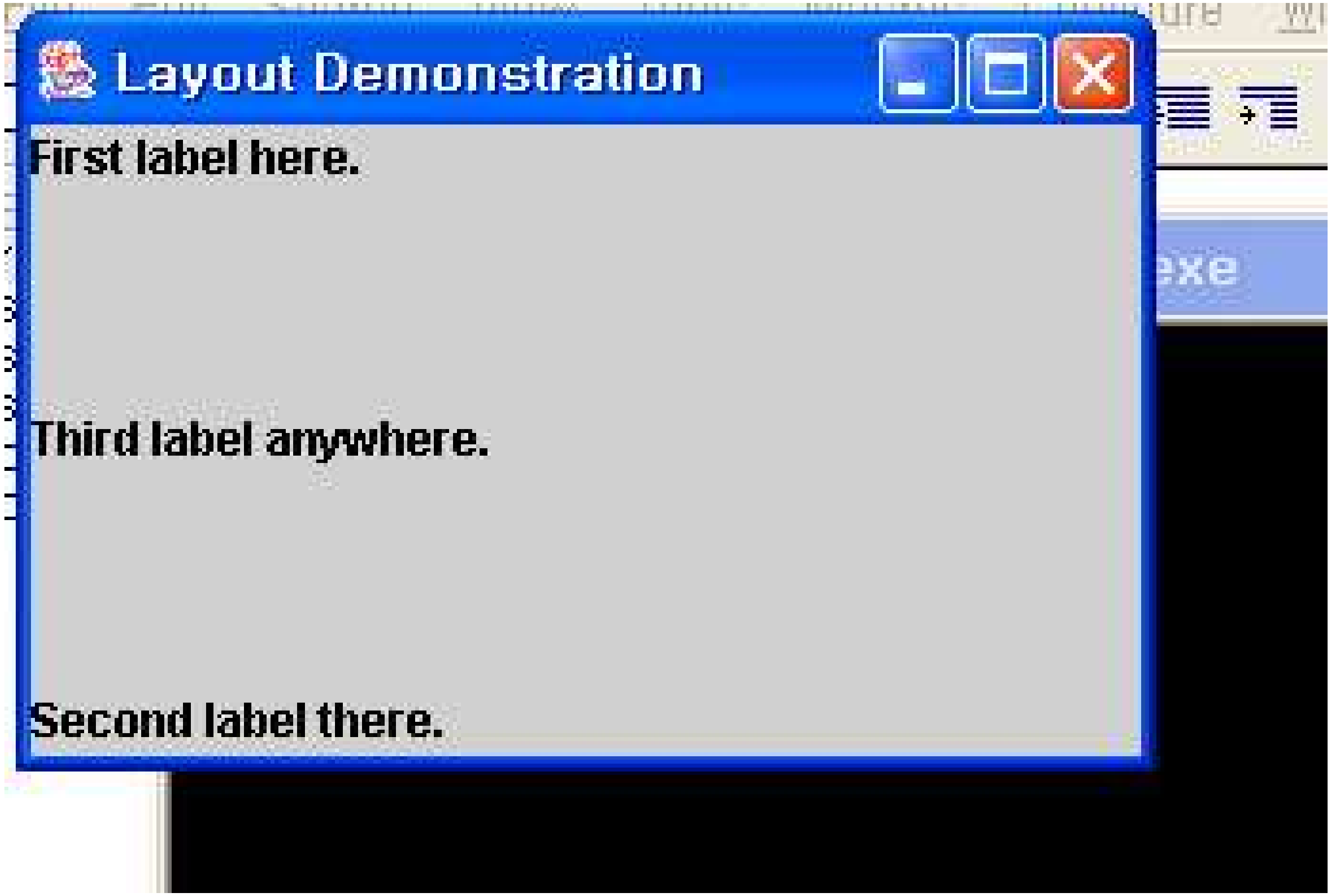
```
  /**
     Creates and displays a window of the class BorderLayoutDemo.
    */
    public static void main(String[] args)
    {
        BorderLayoutDemo gui = new BorderLayoutDemo( );
        gui.setVisible(true);
    }
    public BorderLayoutDemo( )
    {
        setSize(WIDTH, HEIGHT);
        addWindowListener(new WindowDestroyer( ));
        setTitle("Layout Demonstration");
        Container content = getContentPane( );

        content.setLayout(new BorderLayout( ));

        JLabel label1 = new JLabel("First label here.");
        content.add(label1, BorderLayout.NORTH);

        JLabel label2 = new JLabel("Second label there.");
        content.add(label2, BorderLayout.SOUTH);

        JLabel label3 = new JLabel("Third label anywhere.");
        content.add(label3, BorderLayout.CENTER);
    }
}
```



First label here.

Third label anywhere.

Second label there.

The Flow Layout Manager

- The simplest layout manager
- Displays components from left to right in the order they are added to the container
- Add method has one parameter which is the component to add
- FLowLayoutDemo.java

```
Container content = getContentPane();
content.setLayout(new FlowLayout());
JLabel label1 = new JLabel(“First label here”);
content.add(label1);
JLabel label2 = new JLabel(“Second label there”);
content.add(label2);
```

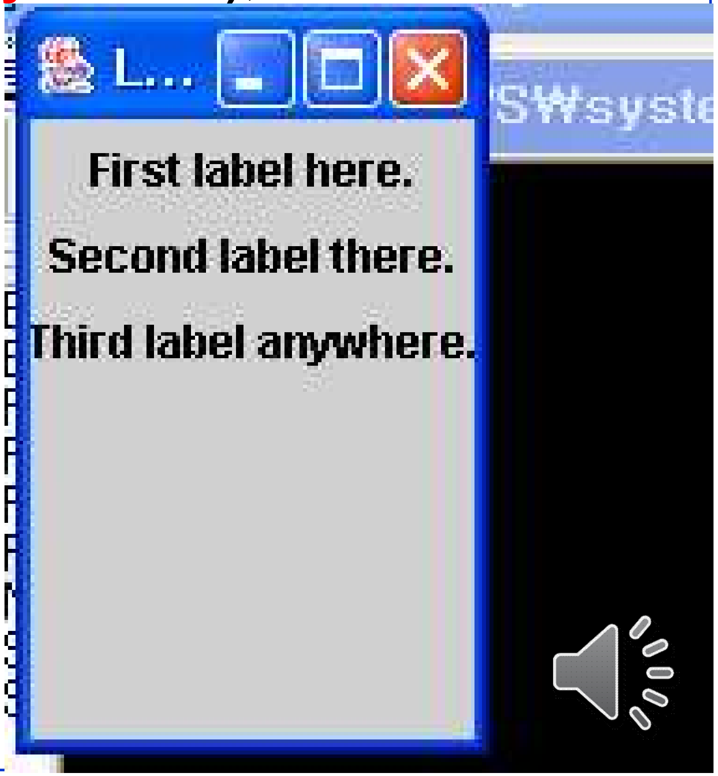
Examples of The Flow Layout Manager - FLowLayoutDemo.java

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

/**
 Simple demonstration of using a layout manager to arrange labels.
*/
public class FlowLayoutDemo extends JFrame
{
    public static final int WIDTH = 100;
    public static final int HEIGHT = 200;

    /**
     Creates and displays a window of the class FlowLayoutDemo.
    */
    public static void main(String[] args)
    {
        FlowLayoutDemo gui = new FlowLayoutDemo();
        gui.setVisible(true);
    }
    public FlowLayoutDemo()
    {
        setSize(WIDTH, HEIGHT);
        addWindowListener(new WindowDestroyer());
        setTitle("Layout Demonstration");
        Container content = getContentPane();

        content.setLayout(new FlowLayout());
        JLabel label1 = new JLabel("First label here.");
        content.add(label1);
        JLabel label2 = new JLabel("Second label there.");
        content.add(label2);
        JLabel label3 = new JLabel("Third label anywhere.");
        content.add(label3);
    }
}
```



First label here.

Second label there.

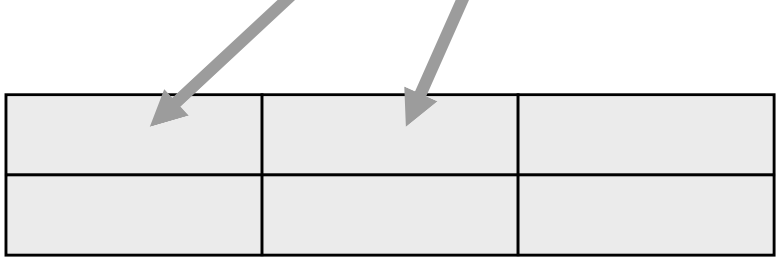
Third label anywhere.

The Grid Layout Manager

- Specify a number of rows and columns
- All regions in the grid are equal size
- When the container changes size, each region grows or shrinks by the same amount
- GridLayoutDemo1.java

```
aContainer.setLayout(new GridLayout(2, 3));
. . .
aContainer.add(label1);
aContainer.add(label2);
```

Creates a grid layout with two rows and three columns.



Rows are filled before columns.

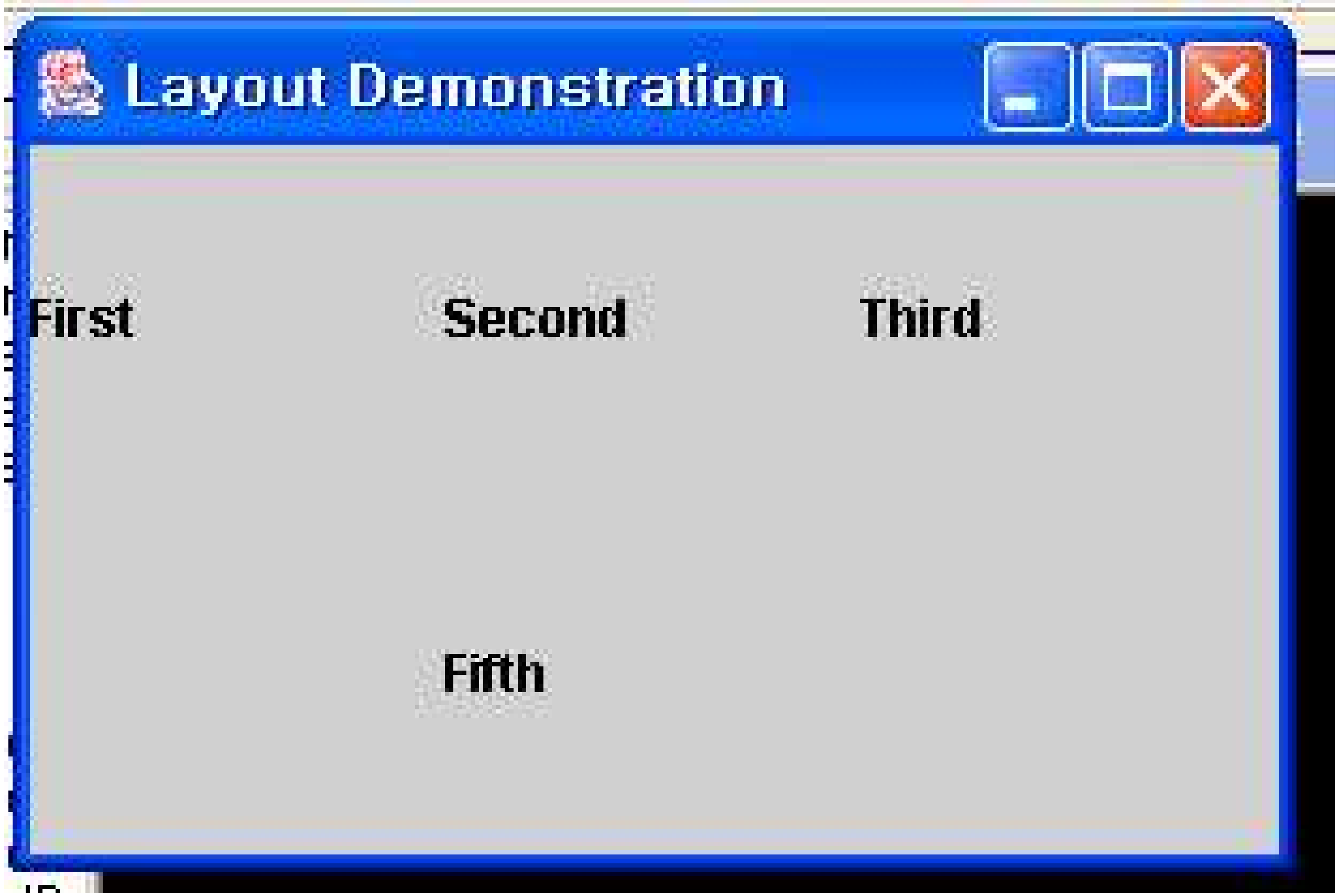
GridLayoutDemo1.java

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

/**
 Simple demonstration of using a layout manager to arrange labels.
*/
public class GridLayoutDemo1 extends JFrame
{
    public static final int WIDTH = 300;
    public static final int HEIGHT = 200;

    /**
     Creates and displays a window of the class BorderLayoutDemo.
    */
    public static void main(String[] args)
    {
        GridLayoutDemo1 gui1 = new GridLayoutDemo1();
        gui1.setVisible(true);
    }
 public GridLayoutDemo1()
  {
    setSize(WIDTH, HEIGHT);
    addWindowListener(new WindowDestroyer());
    setTitle("Layout Demonstration");
    Container content = getContentPane();

    content.setLayout(new GridLayout(2, 3));
    JLabel label1 = new JLabel("First");
    content.add(label1);
    JLabel label2 = new JLabel("Second");
    content.add(label2);
    JLabel label3 = new JLabel("Third");
    content.add(label3);
    JLabel label4 = new JLabel("");//Empty string label
    content.add(label4);
    JLabel label5 = new JLabel("Fifth");
    content.add(label5);
  }
}
```



Layout Demonstration

First

Second

Third

Fifth

The Grid Layout Manager

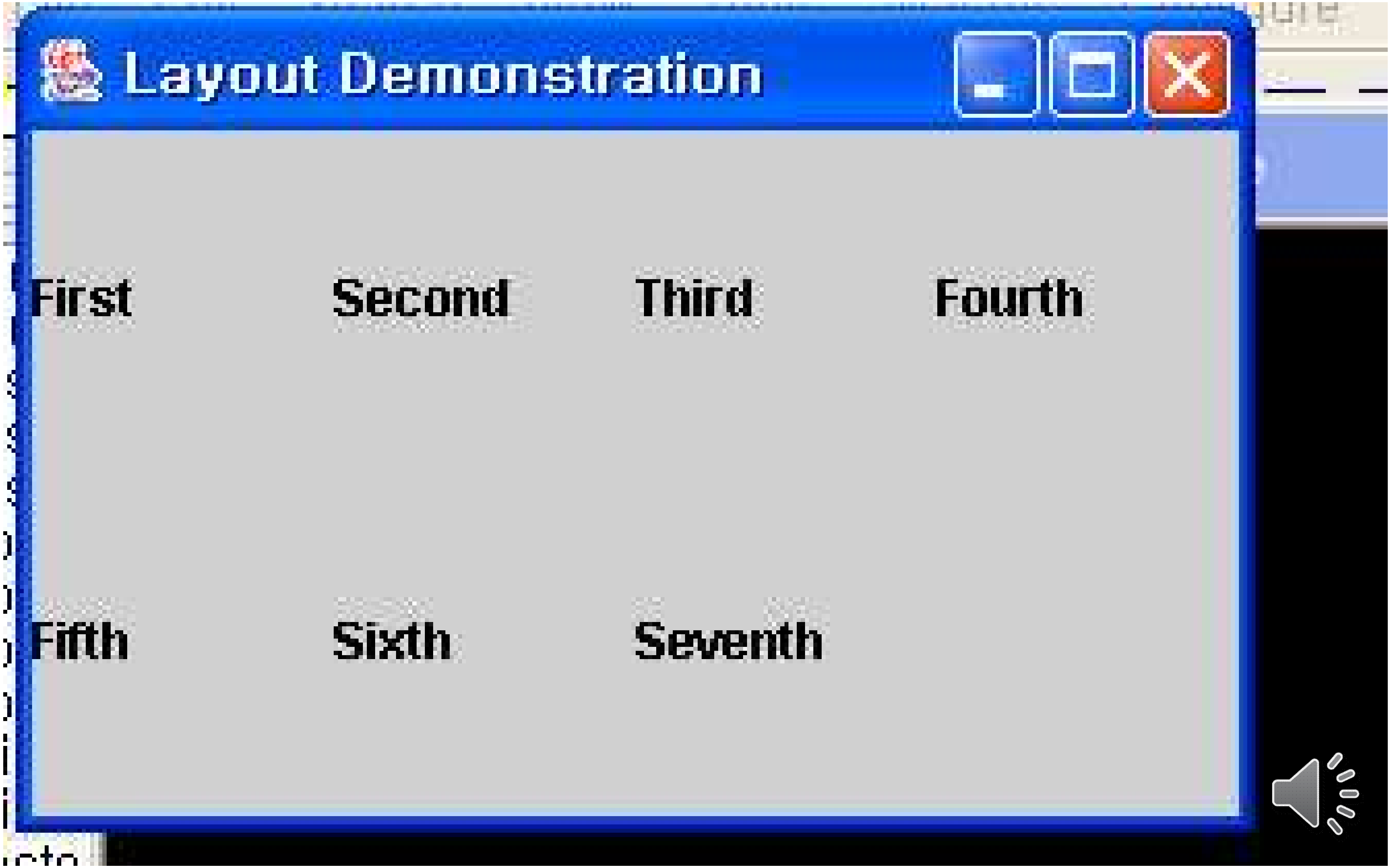
- GridLayoutDemo2.java
  - » The number of columns will actually be determined by the number of items added to the container.
  - » If you add six items, the grid will be as shown
  - » If you add seven or eight items, a fourth column is automatically added.

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

/**
 Simple demonstration of using a layout manager to arrange labels.
 Note that seven items are added even though only six locations are
specified.
*/
public class GridLayoutDemo2 extends JFrame
{
    public static final int WIDTH = 300;
    public static final int HEIGHT = 200;

    /**
     Creates and displays a window of the class BorderLayoutDemo.
    */
    public static void main(String[] args)
    {
        GridLayoutDemo2 gui2 = new GridLayoutDemo2();
        gui2.setVisible(true);
    }
 public GridLayoutDemo2()
  {
    setSize(WIDTH, HEIGHT);
    addWindowListener(new WindowDestroyer());
    setTitle("Layout Demonstration");
    Container content = getContentPane();

    content.setLayout(new GridLayout(2, 3));
    JLabel label1 = new JLabel("First");
    content.add(label1);
    JLabel label2 = new JLabel("Second");
    content.add(label2);
    JLabel label3 = new JLabel("Third");
    content.add(label3);
    JLabel label4 = new JLabel("Fourth");//Empty string label
    content.add(label4);
    JLabel label5 = new JLabel("Fifth");
    content.add(label5);
    JLabel label6 = new JLabel("Sixth");
    content.add(label6);
    JLabel label7 = new JLabel("Seventh");
    content.add(label7);
  }
}
```


Layout Demonstration
First
Second
Third
Fourth
Fifth
Sixth
Seventh

끝