Layout

Chapter Topics

1. What is the Layout class

Back to Chapter Topics

2. The FlowLayout

The FlowLayout class is the most basic of all layouts. The Flowlayout adds components to the Applet panel, along a row, one at a time. Component wrap onto the next row, if they don't fit across. Components can also be aligned (by default the alignment is center). The following code shows how to create a basic flow layout, a flow layout with alignment to the left, and a flow layout with horizontal and vertical margins (or gaps) between components :

```
setLayout(new FlowLayout());

setLayout(new FlowLayout(FlowLayout.LEFT));

setLayout(new FlowLayout(FlowLayout.LEFT), 10, 10);
```

The following example shows how to use the flow layout in an applet with button components.

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

public class flayout extends Applet
{
Button first, second, third, forth;  //This declares the buttons

public void init() {

   setLayout(new FlowLayout());

   first = new Button ("Rewind");  //This initializes the first
   add(first);                    //This adds the button

   second = new Button ("Play");
   add(second);

   third = new Button ("Fast Forward");
   add(third);

   forth = new Button ("Stop");
   add(forth);
}
}
```

View Applet

Back to Chapter Topics

3. The GridLayout

Grid layouts allow programers to create sections within the Applet panel. Components can then be added to the rows and columns of the panel. Each component added to the panel is placed in a cell of the grid, starting from the top row left column through to the bottom row right column. With Grid layouts, the order in which the components are added determines their placement on the applet panel.
The following code shows how to create a basic grid layout as well as a layout with horizontal and vertical margins (or gaps) between components. With grid layouts, you must specify how many rows and columns to be added:

```
setLayout(new GridLayout(3, 3));
```

```
//this layout has 3 rows and 3 columns
setLayout(new GridLayout(2, 3, 10, 15));
//this layout has 2 rows, 3 columns, 10 pixel horizontal margin,
```

The following example shows how to use the grid layout in an applet with button components.

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

public class flayout extends Applet
{
Button first, second, third, forth;  //This declares the buttons

public void init() {

   setLayout(new GridLayout(2,3,10,15));

   first = new Button ("Rewind");  //This initializes the first
   add(first);                    //This adds the button

   second = new Button ("Play");
   add(second);

   third = new Button ("Fast Forward");
   add(third);

   forth = new Button ("Stop");
   add(forth);
}
}
```

View Applet

Back to Chapter Topics

4. The BorderLayout

Border layouts are different from flow and grid layouts in that you must add your component to the specific regions of the layout. Border layout is broken down into north, south, east, west and center. The

components are added to the applet panel and use up as much size as they need.
The following code shows how to create a basic border layout, and a border layout with horizontal and vertical margins (or gaps) between components :

```
setLayout(new BorderLayout());
setLayout(new BorderLayout(10, 10));
```

to add the individual components to the panel, you must specify the specific region as follows:

```
add("North", new TextField("Title", 50));
add("South", new TextField("Status", 50));
```

The following example shows how to use the border layout in an applet with button components and horizontal and vertical margins.

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

public class blayout extends Applet
{
Button first, second, third, forth, fith, sixth;  //This declare

public void init() {

   setLayout(new BorderLayout(10, 10));

   first = new Button ("Rewind");  //This initializes the first
   add("North",first);                //This adds the button

   second = new Button ("Play");
   add("South",second);

   third = new Button ("Fast Forward");
   add("East",third);

   forth = new Button ("Stop");
   add("West",forth);

   fith = new Button ("Pause");
   add("Center",fith);

}
}
```

View Applet

Back to Chapter Topics

5. The Null Layout

The null layout allows programmers to define exactly where to place components on the Applet panel. By setting the layout to null, programmers can then use the applet's coordinate system to place components.
The following code shows how to set the layout to null, and to add components to the applet using setBounds:

```
setLayout(null);
first = new Button ("Rewind");
add(first);
first.setBounds(x1,y1,w,h);
```

The following example shows how to use the border layout in an applet with button components and horizontal and vertical margins.

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

public class nlayout extends Applet
{
Button first, second, third, forth, fith, sixth;  //This declare

public void init() {

   setLayout(null);

   first = new Button ("Play");  //This initializes the first bu
   add(first);
   first.setBounds(20,20,30,20);

   second = new Button ("Rewind");
   add(second);
   second.setBounds(20,50,50,20);
   third = new Button ("Fast Forward");
   add(third);
   third.setBounds(20,80,70,20);

}
}
```

View Applet

Back to Chapter Topics