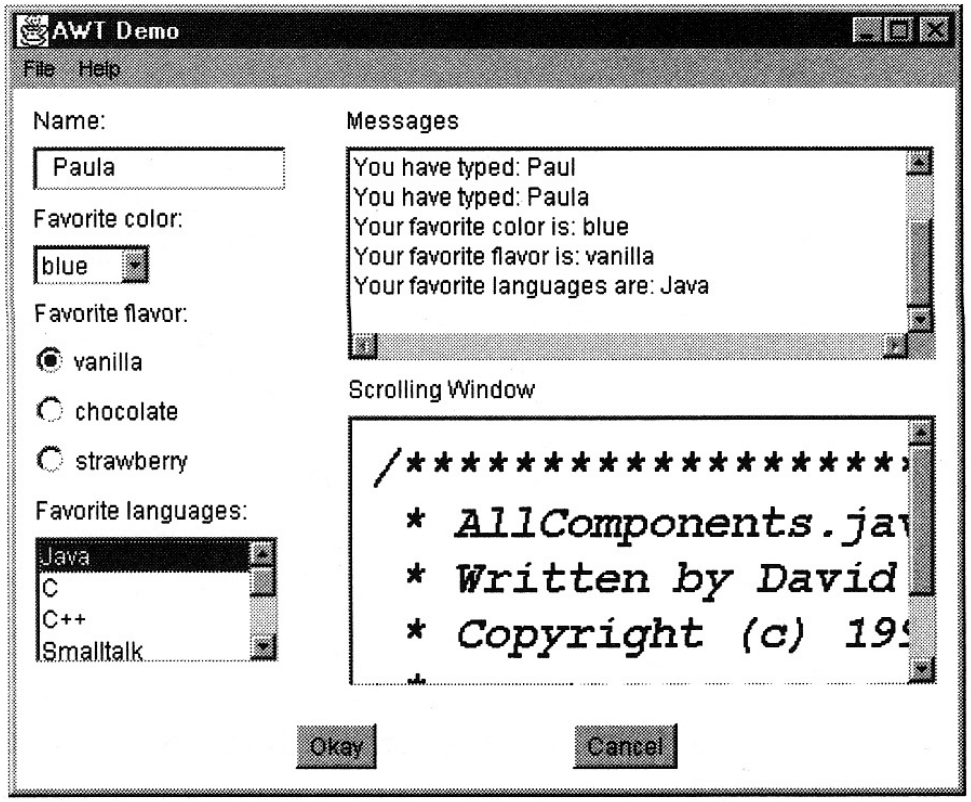
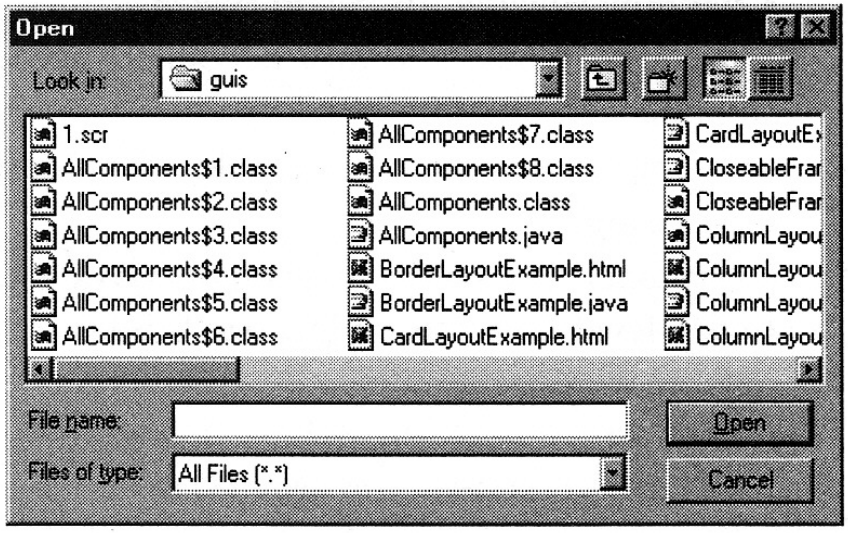
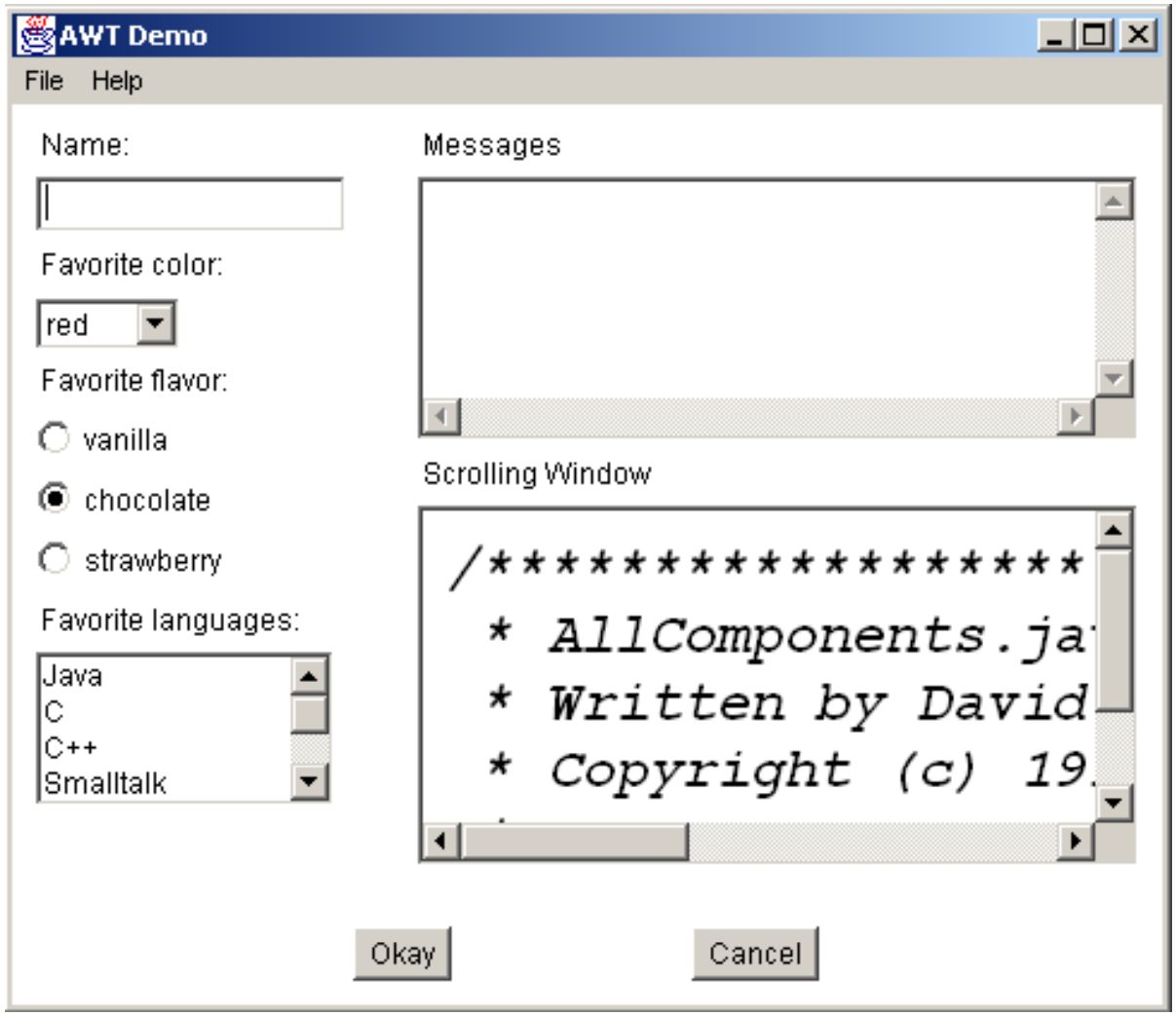
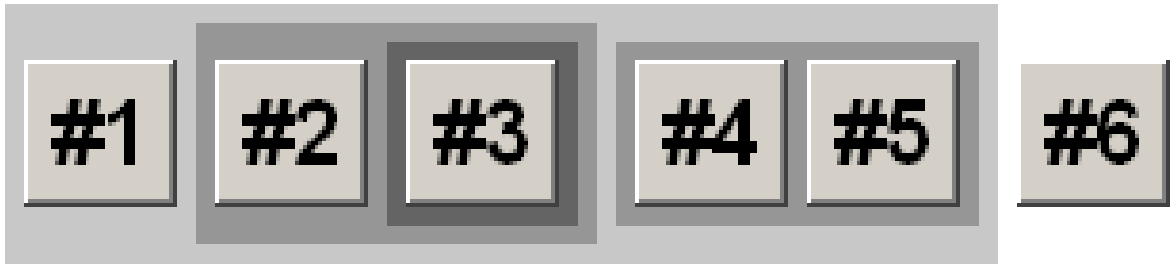
Windows'un Grafik Kullanıcı Arabirimi


AWT Demo
File Help
Name:
Paula
Favorite color:
blue
Favorite flavor:
vanilla
chocolate
strawberry
Favorite languages:
Java
C
C++
Smalltalk
Messages
You have typed: Paul
You have typed: Paula
Your favorite color is: blue
Your favorite flavor is: vanilla
Your favorite languages are: Java
Scrolling Window
/**********
* AllComponents.jav
* Written by David
* Copyright (c) 19
Okay
Cancel


Open
Look in: guis
1.scr
AllComponents$1.class
AllComponents$2.class
AllComponents$3.class
AllComponents$4.class
AllComponents$5.class
AllComponents$6.class
AllComponents$7.class
AllComponents$8.class
AllComponents.class
AllComponents.java
BorderLayoutExample.html
BorderLayoutExample.java
CardLayoutExample.html
CardLayoutEx
CloseableFrar
CloseableFrar
ColumnLayou
ColumnLayou
ColumnLayou
ColumnLayou
File name:
Open
Files of type: All Files (*.*)
Cancel

Graphical User Interfaces

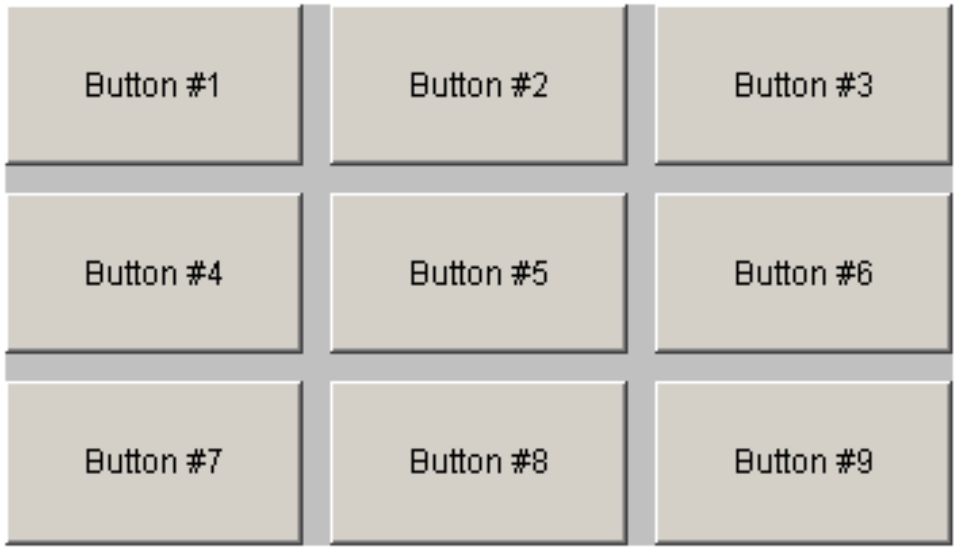




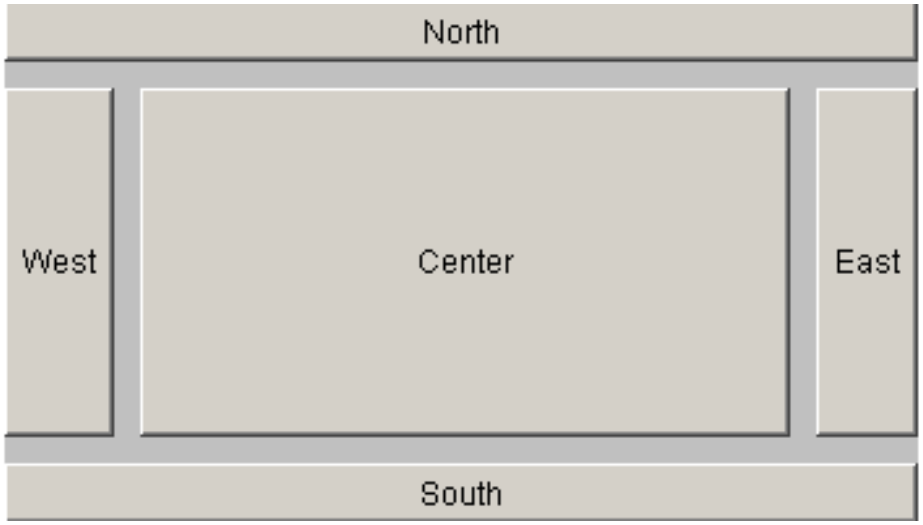

```
import java.applet.*;
import java.awt.*;

/**
 * An applet that demonstrates nested container and components
 * It creates the hierarchy shown below, and uses different colors to
 * distinguish the different nesting levels of the containers
 *
 *   applet---panel1----button1
 *        |       |---panel2----button2
 *        |       |       |----panel3----button3
 *        |       |------panel4----button4
 *        |                 |----button5
 *        |---button6
 */
public class Containers extends Applet {
  public void init() {
    this.setBackground(Color.white);           // The applet is white
    this.setFont(new Font("Dialog", Font.BOLD, 24));
```

```
 Panel p1 = new Panel();
   p1.setBackground(new Color(200, 200, 200)); // Panel1 is darker than applet
   this.add(p1);                 // Panel 1 is contained in applet
   p1.add(new Button("#1"));      // Button 1 is contained in Panel 1

   Panel p2 = new Panel();
   p2.setBackground(new Color(150, 150, 150)); // Panel2 is darker than Panel1
   p1.add(p2);                   // Panel 2 is contained in Panel 1
   p2.add(new Button("#2"));      // Button 2 is contained in Panel 2

   Panel p3 = new Panel();
   p3.setBackground(new Color(100, 100, 100)); // Panel3 is darker than Panel2
   p2.add(p3);                   // Panel 3 is contained in Panel 2
   p3.add(new Button("#3"));      // Button 3 is contained in Panel 3

   Panel p4 = new Panel();
   p4.setBackground(new Color(150, 150, 150)); // Panel4 is darker than Panel1
   p1.add(p4);                   // Panel4 is contained in Panel 1
   p4.add(new Button("#4"));      // Button4 is contained in Panel4
   p4.add(new Button("#5"));      // Button5 is contained in Panel4

   this.add(new Button("#6"));    // Button6 is contained in applet
  }
}
```

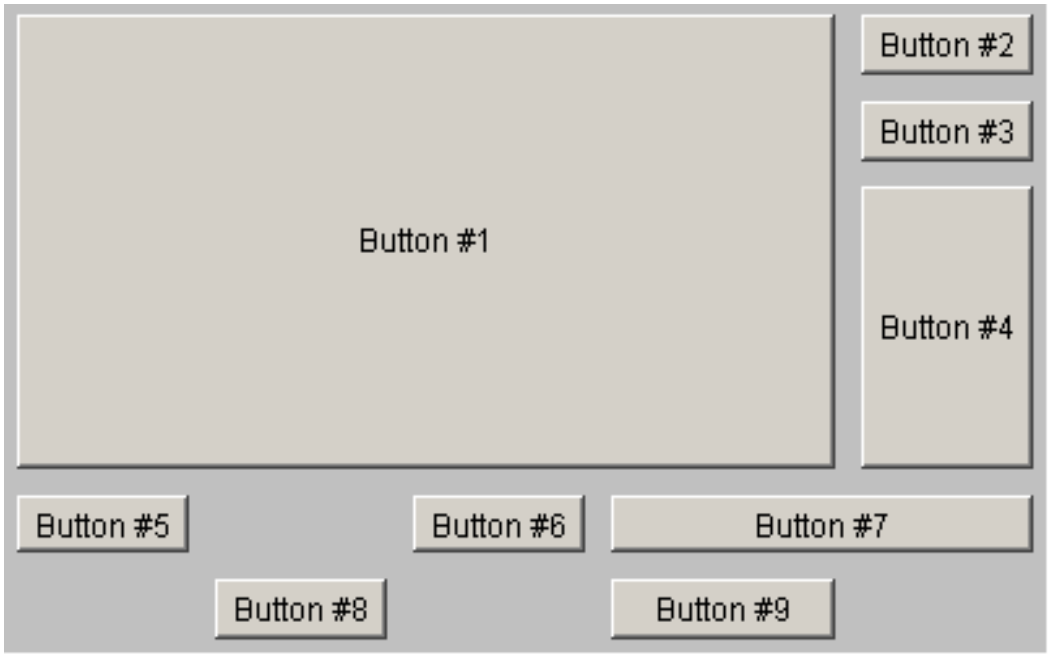

```
import java.applet.*;
import java.awt.*;

public class FlowLayoutExample extends Applet {
  public void init() {
    // Create and specify the layout manager for the applet container.
    // Leave 10 pixels of horizontal and vertical space between components.
    // Left justify rows.
    this.setLayout(new FlowLayout(FlowLayout.LEFT, 10, 10));
    String spaces = "";  // Used to make the buttons different sizes
    for(int i = 1; i <= 9; i++) {
      this.add(new Button("Button #" + i + spaces));
      spaces += " ";
    }
  }
}
```




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

public class GridLayoutExample extends Applet {
  public void init() {
    // Create and specify a layout manager for this applet.
    // Layout components into a grid three columns wide, with the number
    // of rows to depend on the number of components.  Leave 10 pixels
    // of horizontal and vertical space between components
    this.setLayout(new GridLayout(0, 3, 10, 10));
    for(int i = 1; i <= 9; i++)
      this.add(new Button("Button #" + i));
  }
}
```



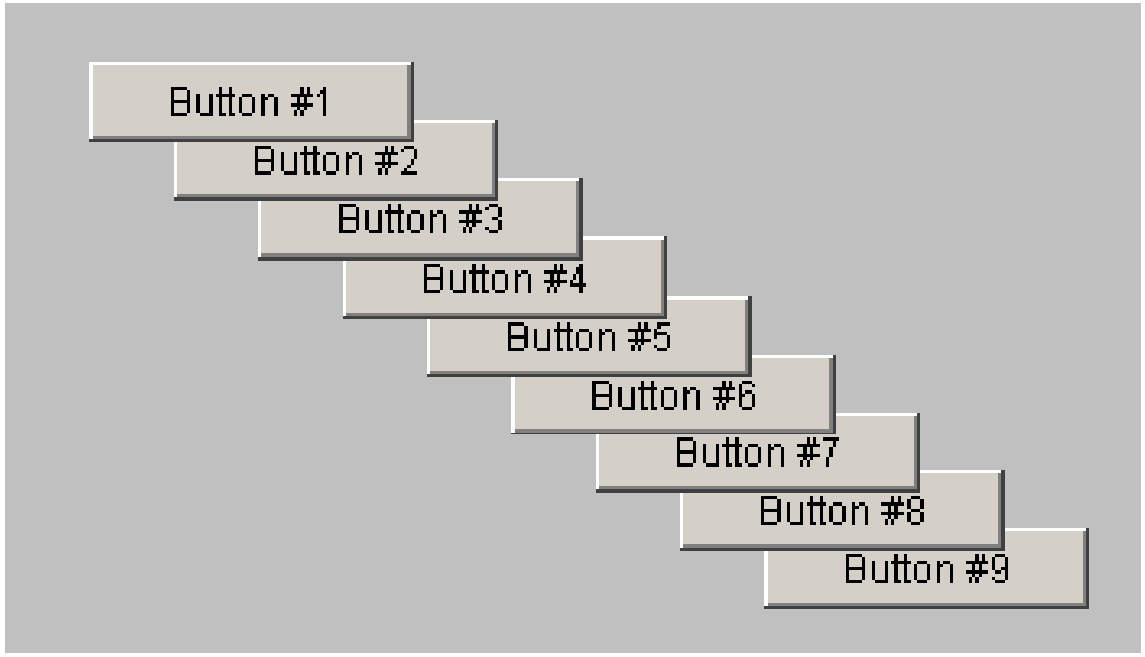

```
import java.applet.*;
import java.awt.*;

public class BorderLayoutExample extends Applet {
  String[] borders = {"North", "East", "South", "West", "Center"};
  public void init() {
    // Create and specify a BorderLayout layout manager that leaves
    // 10 pixels of horizontal and vertical space between components
    this.setLayout(new BorderLayout(10, 10));
    for(int i = 0; i < 5; i++) {
      // Swap the order of these arguments in Java 1.1
      this.add(borders[i], new Button(borders[i]));
    }
  }
}
```




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

public class GridBagLayoutExample extends Applet {
  public void init() {
    // Create and specify a layout manager
    this.setLayout(new GridBagLayout());

    // Create a constraints object, and specify some default values
    GridBagConstraints c = new GridBagConstraints();
    c.fill = GridBagConstraints.BOTH;  // components grow in both dimensions
    c.insets = new Insets(5,5,5,5);    // 5-pixel margins on all sides
```

```
// Create and add a bunch of buttons, specifying different grid
  // position, and size for each.
  // Give the first button a resize weight of 1.0 and all others
  // a weight of 0.0.  The first button will get all extra space.
  c.gridx = 0; c.gridy = 0; c.gridwidth = 4; c.gridheight=4;
  c.weightx = c.weighty = 1.0;
  this.add(new Button("Button #1"), c);

  c.gridx = 4; c.gridy = 0; c.gridwidth = 1; c.gridheight=1;
  c.weightx = c.weighty = 0.0;
  this.add(new Button("Button #2"), c);

  c.gridx = 4; c.gridy = 1; c.gridwidth = 1; c.gridheight=1;
  this.add(new Button("Button #3"), c);

  c.gridx = 4; c.gridy = 2; c.gridwidth = 1; c.gridheight=2;
  this.add(new Button("Button #4"), c);

  c.gridx = 0; c.gridy = 4; c.gridwidth = 1; c.gridheight=1;
  this.add(new Button("Button #5"), c);

 c.gridx = 2; c.gridy = 4; c.gridwidth = 1; c.gridheight=1;
  this.add(new Button("Button #6"), c);

  c.gridx = 3; c.gridy = 4; c.gridwidth = 2; c.gridheight=1;
  this.add(new Button("Button #7"), c);
```

```
 c.gridx = 1; c.gridy = 5; c.gridwidth = 1; c.gridheight=1;
    this.add(new Button("Button #8"), c);

    c.gridx = 3; c.gridy = 5; c.gridwidth = 1; c.gridheight=1;
    this.add(new Button("Button #9"), c);
  }
}
```



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

public class NullLayoutExample extends Applet {
  public void init() {
    // Get rid of the default layout manager.
    // We'll arrange the components ourself.
    this.setLayout(null);
    for(int i = 1; i <= 9; i++) {
      Button b = new Button("Button #" + i);
      b.setBounds(i*26, i*18, 100, 25); // use reshape() in Java 1.0
      this.add(b);
    }
  }
  public Dimension getPreferredSize() { return new Dimension(350, 225); }
}
```

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

/** A program that uses all the standard AWT components */
public class AllComponents extends Frame implements ActionListener {
  TextArea textarea;  // Events messages will be displayed here.

  /** Create the whole GUI, and set up event listeners */
  public AllComponents(String title) {
    super(title);  // set frame title.

    // Arrange to detect window close events
    this.addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e) { System.exit(0); }
    });

    // Set a default font
    this.setFont(new Font("SansSerif", Font.PLAIN, 12));

    // Create the menubar.  Tell the frame about it.
    MenuBar menubar = new MenuBar();
    this.setMenuBar(menubar);

    // Create the file menu.  Add to menubar.
    Menu file = new Menu("File");
    menubar.add(file);
```

```
// Create two items for the file menu, setting their label, shortcut,
  // action command and listener.  Add them to File menu.
  // Note that we use the frame itself as the action listener
  MenuItem open = new MenuItem("Open", new MenuShortcut(KeyEvent.VK_O));
  open.setActionCommand("open");
  open.addActionListener(this);
  file.add(open);
  MenuItem quit = new MenuItem("Quit", new MenuShortcut(KeyEvent.VK_Q));
  quit.setActionCommand("quit");
  quit.addActionListener(this);
  file.add(quit);

  // Create Help menu; add an item; add to menubar
  // Display the help menu in a special reserved place.
  Menu help = new Menu("Help");
  menubar.add(help);
  menubar.setHelpMenu(help);

  // Create and add an item to the Help menu
  MenuItem about = new MenuItem("About", new MenuShortcut(KeyEvent.VK_A));
  about.setActionCommand("about");
  about.addActionListener(this);
  help.add(about);
```

```
// Now that we've done the menu, we can begin work on the contents of
  // the frame.  Assign a BorderLayout manager with margins for this frame.
  this.setLayout(new BorderLayout(10, 10));

  // Create two panels to contain two columns of components.  Use our custom
  // ColumnLayout layout manager for each.  Add them on the west and
  // center of the frame's border layout
  Panel column1 = new Panel();
  column1.setLayout(new ColumnLayout(5, 10, 2, ColumnLayout.LEFT));
  this.add(column1, "West");
  Panel column2 = new Panel();
  column2.setLayout(new ColumnLayout(5, 10, 2, ColumnLayout.LEFT));
  this.add(column2, "Center");

  // Create a panel to contain the buttons at the bottom of the window
  // Give it a FlowLayout layout manager, and add it along the south border
  Panel buttonbox = new Panel();
  buttonbox.setLayout(new FlowLayout(FlowLayout.CENTER, 100, 10));
  this.add(buttonbox, "South");

  // Create pushbuttons and add them to the buttonbox
  Button okay = new Button("Okay");
  Button cancel = new Button("Cancel");
  buttonbox.add(okay);
  buttonbox.add(cancel);
```

```
// Handle events on the buttons
  ActionListener buttonlistener = new ActionListener() {
    public void actionPerformed(ActionEvent e) {
      textarea.append("You clicked: " +
                      ((Button)e.getSource()).getLabel() + "\n");
    }
  };
  okay.addActionListener(buttonlistener);
  cancel.addActionListener(buttonlistener);

  // Now start filling the left column.
  // Create a 1-line text field and add to left column, with a label
  TextField textfield = new TextField(15);
  column1.add(new Label("Name:"));
  column1.add(textfield);

  // Handle events on the TextField
  textfield.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
      textarea.append("Your name is: " +
                      ((TextField)e.getSource()).getText() + "\n");
    }
  });
  textfield.addTextListener(new TextListener() {
    public void textValueChanged(TextEvent e) {
      textarea.append("You have typed: " +
                      ((TextField)e.getSource()).getText() + "\n");
    }
  });
```

```
// Create a dropdown list or option menu of choices
  Choice choice = new Choice();
  choice.addItem("red");
  choice.addItem("green");
  choice.addItem("blue");
  column1.add(new Label("Favorite color:"));
  column1.add(choice);

  // Handle events on this choice
  choice.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
      textarea.append("Your favorite color is: " + e.getItem() + "\n");
    }
  });

  // Create checkboxes, and group them in a CheckboxGroup to give them
  // "radio button" behavior.
  CheckboxGroup checkbox_group = new CheckboxGroup();
  Checkbox[] checkboxes = new Checkbox[3];
  checkboxes[0] = new Checkbox("vanilla", checkbox_group, false);
  checkboxes[1] = new Checkbox("chocolate", checkbox_group, true);
  checkboxes[2] = new Checkbox("strawberry", checkbox_group, false);
  column1.add(new Label("Favorite flavor:"));
```

```
for(int i = 0; i < checkboxes.length; i++) column1.add(checkboxes[i]);
  // Handle events on the checkboxes
  ItemListener checkbox_listener = new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
      textarea.append("Your favorite flavor is: " +
                      ((Checkbox)e.getItemSelectable()).getLabel() + "\n");
    }
  };
  for(int i = 0; i < checkboxes.length; i++)
    checkboxes[i].addItemListener(checkbox_listener);
  // Create a list of choices.
  List list = new List(4, true);
  list.addItem("Java"); list.addItem("C"); list.addItem("C++");
  list.addItem("Smalltalk"); list.addItem("Lisp");
  list.addItem("Modula-3"); list.addItem("Forth");
  column1.add(new Label("Favorite languages:"));
  column1.add(list);
  // Handle events on this list
  list.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
      textarea.append("Your favorite languages are: ");
      String[] languages = ((List)e.getItemSelectable()).getSelectedItems();
      for(int i = 0; i < languages.length; i++) {
        if (i > 0) textarea.append(",");
        textarea.append(languages[i]);
      }
      textarea.append("\n");
    }
  });
```

```
// Create a multi-line text area in column 2
  textarea = new TextArea(6, 40);
  textarea.setEditable(false);
  column2.add(new Label("Messages"));
  column2.add(textarea);

  // Create a scrollpane that displays portions of a larger component
  ScrollPane scrollpane = new ScrollPane();
  scrollpane.setSize(300, 150);
  column2.add(new Label("Scrolling Window"));
  column2.add(scrollpane);

  // Create a custom MultiLineLabel with a really big font and make it
  // a child of the ScrollPane container
  String message =
    "/**********************************************\n" +
    " * AllComponents.java                         *\n" +
    " * Written by David Flanagan                  *\n" +
    " * Copyright (c) 1997 by O'Reilly & Associates   *\n" +
    " *                                            *\n" +
    " **********************************************/\n";
  MultiLineLabel biglabel =  new MultiLineLabel(message);
  biglabel.setFont(new Font("Monospaced", Font.BOLD + Font.ITALIC, 24));
  scrollpane.add(biglabel);
 }
```

```
/** This is the action listener method that the menu items invoke */
 public void actionPerformed(ActionEvent e) {
  String command = e.getActionCommand();
  if (command.equals("quit")) {
    YesNoDialog d = new YesNoDialog(this, "Really Quit?",
                                    "Are you sure you want to quit?",
                                    "Yes", "No", null);
    d.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals("yes")) System.exit(0);
        else textarea.append("Quit not confirmed\n");
      }
    });
    d.show();
  }
  else if (command.equals("open")) {
    FileDialog d = new FileDialog(this, "Open File", FileDialog.LOAD);
    d.show();  // display the dialog and block until answered
    textarea.append("You selected file: " + d.getFile() + "\n");
    d.dispose();
  }
```

```
 else if (command.equals("about")) {
     InfoDialog d = new InfoDialog(this, "About",
                         "This demo was written by David Flanagan\n" +
                         "Copyright (c) 1997 O'Reilly & Associates");
     d.show();
    }
  }

  public static void main(String[] args) {
    Frame f = new AllComponents("AWT Demo");
    f.pack();
    f.show();
  }
}
```

```
    f.setSize(400, 400);     // Set the size of the window
    f.show();                // Finally, pop the window up.
```