Applets

and

Applications

- Application
- Applet
- HTML for applets
- class Applet
- Inevitable HelloWorld example
- More simple applet examples

Applets and Applications

- Application
  - independent program (maybe fully compiled, though more likely byte-code compiled and then interpreted)
  - full access to host machine
    - normal file access (standard security constraints on file ownership)
    - ability to create sockets (arbitrary network connections)
    - (slightly restricted) access to environment variables
  - GUI
    - create Frame object as principal window
    - may have additional windows for dialogs, alerts etc

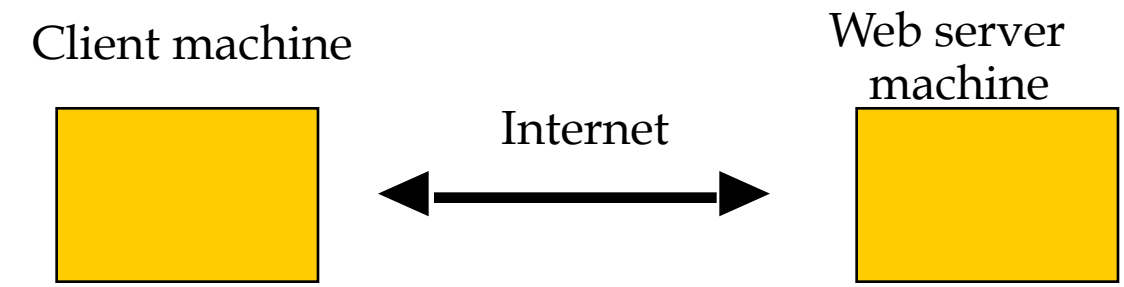
Applets and Applications

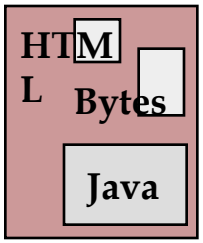
- Applet
  - program embedded in web page and run by "browser"
  - restricted access to machine on which it is run
    (restrictions can be varied by sophisticated browser)
    - no access to local files
    - can open network connection only to site from where applet was itself loaded
    - can access some details of enclosing web page
  - GUI
    - use part of browser's page window for main window

Applet

- Applets do have many restrictions
  - but they were the feature that popularised Java

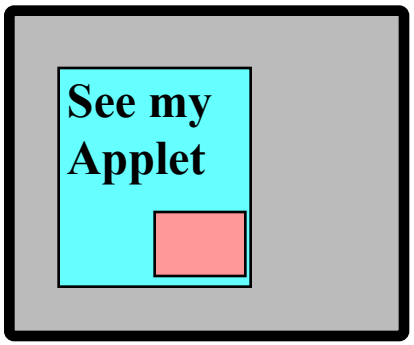
  - ***downloadable executable content***
  - ***security policy easily enforced by browser environment***
  - ***lots more pretty GUI features than you could readily obtain with just HTML+forms***
  - ***real computations, not just checks on data entry & trivial calculations as in Javascript***

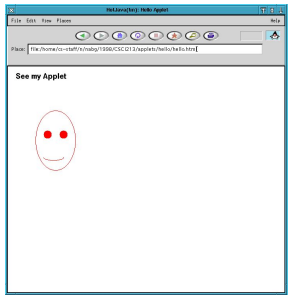




Web browser program
(Java byte interpreter, HTML, applet byte codes)

httpd web server program

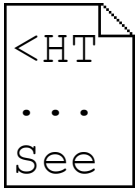




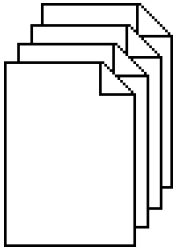
Screen with browser window containing Applet subwindow

Server file system

/http_docs



hello.html



myapplet.class
mycanvas.class
datathing.class
...

- Application
- Applet
- HTML for applets
- class Applet
- Inevitable HelloWorld example
- More simple applet examples

Applet -HTML

- HTML "code" on web page contains tag specifying applet (**<applet ...**
  - somewhat like a **<a href ... >** link, contains name of file with code for applet class, (and dimensions of subwindow required)
  - optionally contains
    - specification of position of window (relative to surrounding text)
    - ...
  - associated with <applet ...> tag may have parameter tags that provide "command line" and "environment" data

Applet - loading & starting

- Browser reads HTML text
  - when encounters "**applet**" tag
    - open connection to server
    - fetch byte codes (code for "myapplet.class")
    - byte codes passed to Java interpreter where get validated by class loader
  - when page loading complete
    - browser starts Java interpreter
    - interpreter opens additional connections to fetch other classes, each loaded by class loader
    - applet object starts to run

Using Applets

- As well as writing .java files with source, you must prepare a HTML file.
- You have to run via a browser
  - AppletViewer
    - simple, fast to load, ignores all content of HTML file except the applet (best for testing)
  - HotJava
    - interprets HTML, so can see applet in context of page as intended for final users
  - IE4, Communicator
    - some bugs re Java1.1

HTML for Applets

<html> <head>
<title>Applet demo</title> </head>
<body>
See my applet
<applet code="MyApplet.class"
... ...
>
Your browser is Java challenged, modernise!
</applet>
</body> </html>

Text between <applet ... > and </applet> displayed if browser doesn't support applets.

HTML for Applets
options in <applet

```
<applet code="MyApplet.class"
   width=200 height=120
   align=
  alt="someone has disabled Java in your browser"
  name=
  codebase=
  archive=
>
```

- **align** left, right, bottom, top, texttop, baseline ...
- **alt** (rare, catering for situation of Java aware browser with Java deactivated)

HTML for Applets options in <applet

- name
  - a page can contain more than one applet;
  - the different applets on a particular page can communicate (rare to want this, Horstmann gives example on page 495);
  - an applet's "name" is used (by another applet on same page) to get a reference object for communication.

options in <applet

- code and codebase
  - **code** name of file with applet class
  - **codebase**
    - normal to organize directories as follows
      - level 1, the .html page(s)
      - level 2,
        - images subdirectory (any pictures used)
        - applet1 subdirectory
        - applet2 subdirectory
        - ...
    - subdirectory identified by codebase

<applet code="MyApplet.class" codebase="myapplet"

myapplet/MyApplet.class

options in <applet

- archive (*more on archives and JAR files later*)
  - pack many classes into an "archive" file
  - saves downloading time,
    - one connection to server, several classes fetched
  - archive argument allows you to specify name(s) of archive file(s)

Parameters for applet

- Parameters are (name, value) pairs; both name and value are strings.
- Uses are generally similar to environment variables (or command line arguments) for Unix programs
  - example, applet that displays pictures and plays back sound; same applet on several pages, with parameters specifying different picture and sound files
  - example, generated HTML page with parameters used to pass current data (eg exchange rate)

Parameters for applet

<applet code="picdisplay.class" width=500 height=300>

<param name="picfile" **value**="images/cat.gif">

<param name="picheight" value="100">

<param name="picwidth" value="150">

you have a java challenged browser, modernise

</applet>

- params placed between **<applet ...>** and **</applet>** tags

- Application
- Applet
- HTML for applets
- class Applet
- Inevitable HelloWorld example
- More simple applet examples

class Applet

class MyApplet extends Applet { ... }

An Applet is a

Panel is a

Container is a

Component is an

Object

Application

class MyApplication *extends Frame* { ... }

An "application" (with a GUI) is something that uses (or, sometimes, *is*) a Frame which is a

Window is a

Container is a

Component is an

Object

class Applet

- Because an Applet is a "**Component**"
  - it is a basic GUI element in its own right
  - it has a paint() method, a repaint(), an update(), ...
- Because it is also a "**Container**"
  - it can have "subwindows", so can hold Canvases, Checkboxes, Buttons, ...
- And, because it is a "**Panel**", it has a defined way of arranging "subwindows"

Applet - execution control (these may be overridden in subclasses)

class Applet

- An Applet has its own unique functionality
  - init() first method called when applet being started; typically does things like read parameters, load images, ...
  - start() called after init() and each time user returns to HTML page with applet; typical use - starting "threads", ...
  - stop() called each time user leaves HTML page; typical use - suspend "threads"
  - destroy() free resources (doesn't Java do this? well, there are problems, particularly on PCs, with Graphics objects) and kill threads

Applet - access "environment"

class Applet

- An Applet has its own unique functionality
  - getAppletContext() returns an object which allows some communication with browser & HTML page
  - getParameter(String name) returns "value" of parameter with "name" (or null)
  - getCodeBase(), getDocumentBase() return URLs of page, etc
  - showStatus(String msg) puts message in "status window" of browser

Applet - manipulate multimedia data

class Applet

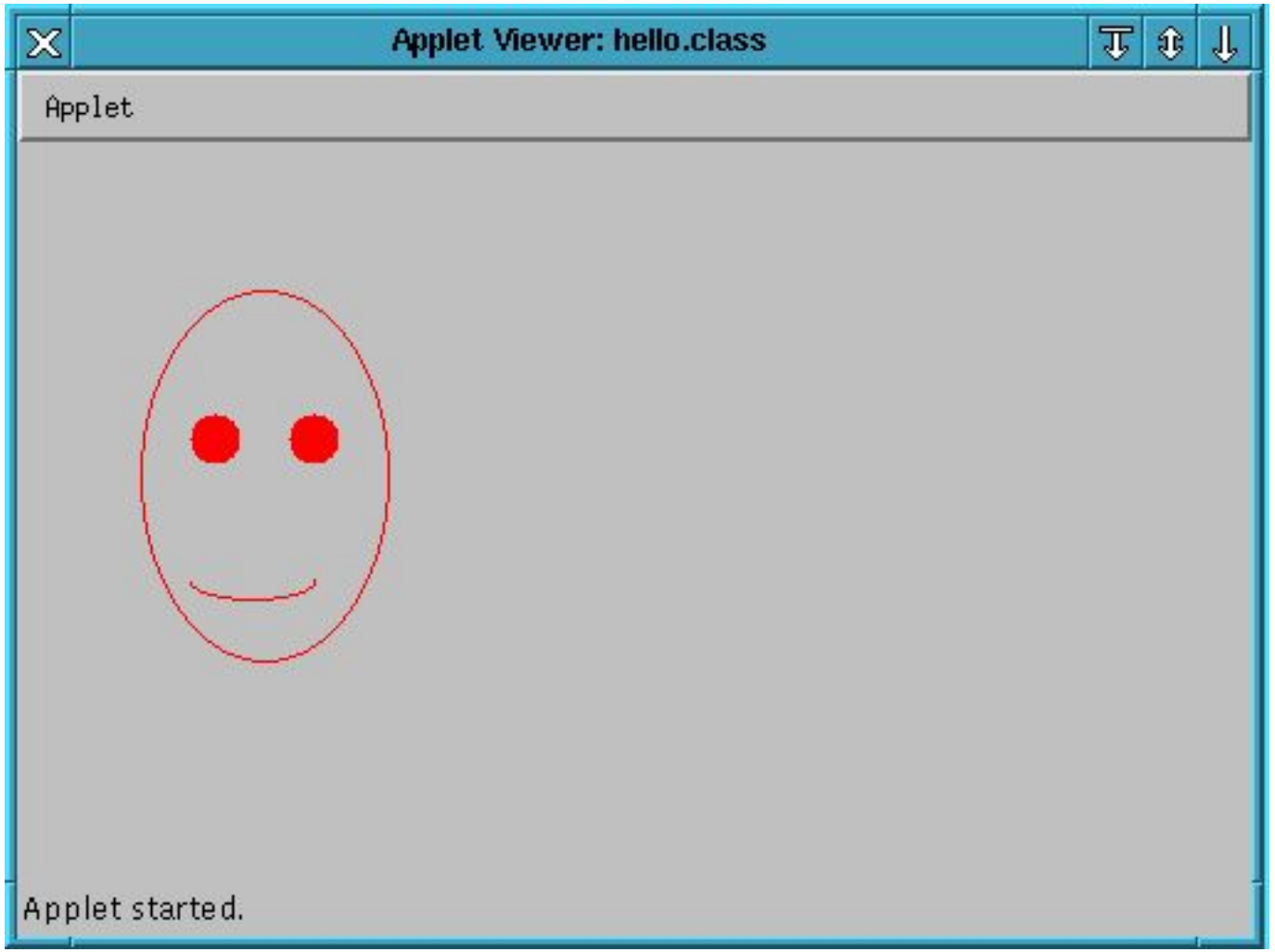
- An Applet has its own unique functionality
  - getImage(URL src) arranges to load an image (eg a .gif file) that is to be displayed in (a subwindow of) Applet
  - getAudioClip(URL src) arranges to load a sound file
  - play(URL src) plays sound file

- Image not loaded until try to display, then may get delays.
- Loading an Audio allows finer control, can start sound clip, stop it etc; play() simply plays it, or does nothing if cannot load.

- Multimedia features figure prominantly in toy applets.

- Application
- Applet
- HTML for applets
- class Applet
- Inevitable HelloWorld example
- More simple applet examples

The inevitable HelloWorld applet


Applet Viewer: hello.class
Applet
Applet started.

HelloWorld applet

- All it does is
  - pick a color using <param> argument
  - draw in its “subwindow” within browser
- So,
  - a specialised subclass of Applet
    - has an init() method that reads parameter
    - has a paint(Graphics g) method that draws
    - accepts defaults (do nothing) for start(), stop(), ...

HTML file for Hello Applet

```
<html> <head>
<title>Hello Applet</title>
</head>
<body>
<h1>See my Applet</h1>
<applet code="hello.class" width=500 height=300>
<param name="color" value="red">
You are Java challenged!
</applet>
</body> </html>
```

class hello *extends Applet*

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

public class hello extends Applet {
    Color c = Color.black;
    public void init() {  ... }
    public void paint(Graphics g) { ... }
}
```

***public* class is required, even if only class in file**

class hello *extends Applet*

```
public void init() {
    String colorval = getParameter("color");
    if(colorval == null) return;
    if(colorval.equals("red")) c= Color.red;
    if(colorval.equals("blue")) c= Color.blue;
}
```

class hello *extends Applet*

```
public void paint(Graphics g) {
    g.setColor(c);
    g.drawOval(50, 60, 100, 150);
    g.fillOval(70, 110, 20, 20);
    g.fillOval(110, 110, 20, 20);
    g.drawArc(70, 170, 50, 15, 0, -180);
    }
```

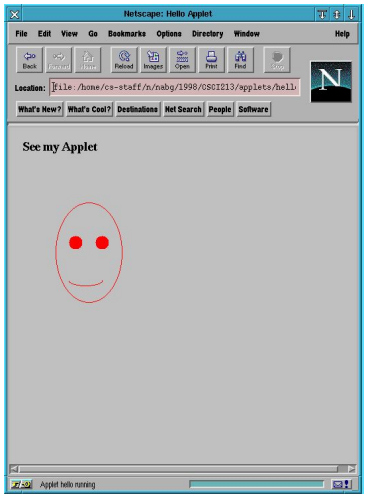
- create hello.html and hello.java
- javac hello.java
- appletviewer hello.html

or

- netscape, open hello.html

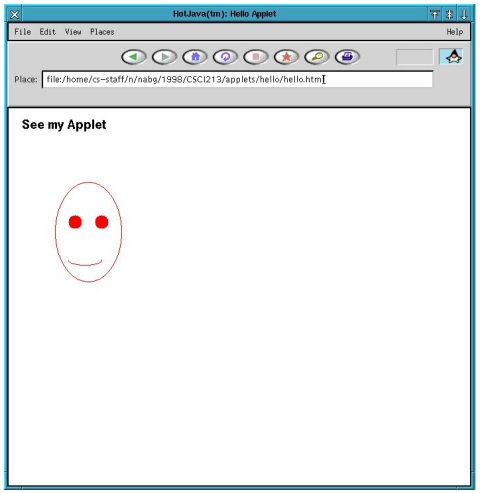
or

- hotjava, open hello.html



Normally, don't get any border around applet subwindow

Best use hotjava or appletviewer;



- Application
- Applet
- HTML for applets
- class Applet
- Inevitable HelloWorld example
- More simple applet examples

More Applet examples

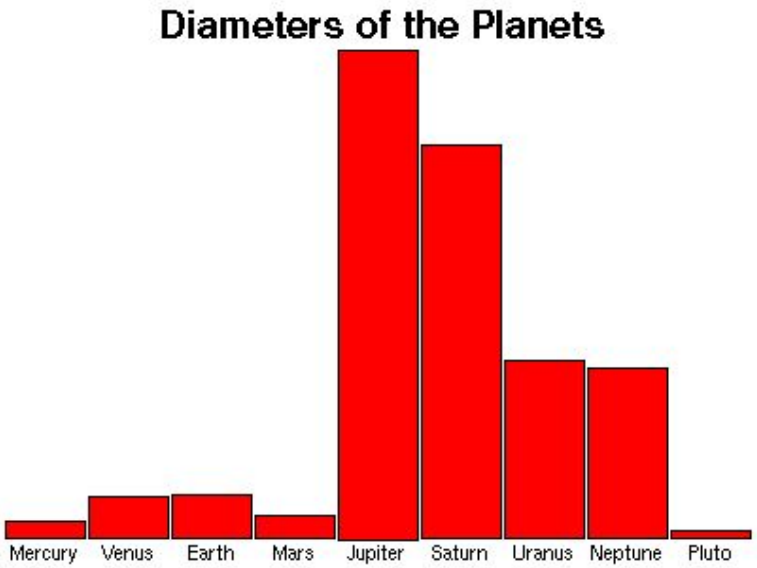
Horstmann's example applets

- Following examples from Horstmann, Core Java
  - chart applet
  - bookmark applet
- also
  - use of a simple button and dialog box
- *(No multimedia examples, they require use of web-accesible directories - which you aren't allowed on your Unix accounts)*

HotJava(tm)

File Edit View Places Help

Place: file:/home/cs-staff/n/nabg/1998/CSCI213/horstmann/CoreJavaBook/ch8/Chart/Chart.htm



Really nothing much more than the "HelloWorld" example

Horstmann's Chart Applet

- Illustration of use of <param ...>
- Applet
  - draw bar chart
    - named columns
    - relative heights defined by double values
- <param...> tags allow same applet to display different data on different pages
  - could use in generated page with <param ...> tags as output from some program

Minor mods to use standard i/o rather than Horstmann's library

```
<APPLET CODE="Chart.class" WIDTH=400 HEIGHT=300>
<PARAM NAME="title" VALUE="Diameters of the Planets">
<PARAM NAME="values" VALUE="9">
<PARAM NAME="name_1" VALUE="Mercury">
<PARAM NAME="name_2" VALUE="Venus">
...
<PARAM NAME="name_9" VALUE="Pluto">
<PARAM NAME="value_1" VALUE="3100">
<PARAM NAME="value_2" VALUE="7500">
<PARAM NAME="value_3" VALUE="8000">
<PARAM NAME="value_4" VALUE="4200">
...
<PARAM NAME="value_9" VALUE="1430">
</APPLET>
```

Chart.html

Chart.java

```
import java.awt.*;
import java.applet.*;
import java.io.*;
public class Chart extends Applet
{
    public void init()   { ... }
   public void paint(Graphics g) {  ... }
   private double[ ] values;
   private String[ ] names;
   private String title;
}
```

```
public void init()
{  int n =  Integer.parseInt(getParameter("values").trim());
     values = new double[n];
     names = new String[n];
     title = getParameter("title");
     int i;
     for (i = 0; i < n; i++)
     {  String s = getParameter("value_" + (i + 1));
        values[i]
           = Double.valueOf(s.trim()).doubleValue();
        names[i] = getParameter("name_" + (i + 1));
     }
  }
```

Note "standard" approach to dealing with variable number of <param ... > arguments

```
public void paint(Graphics g){
    // find range of values to be plotted
    // select font, display title
    // choose scale for plotting items
   // loop to draw each item, and place item label
}
```

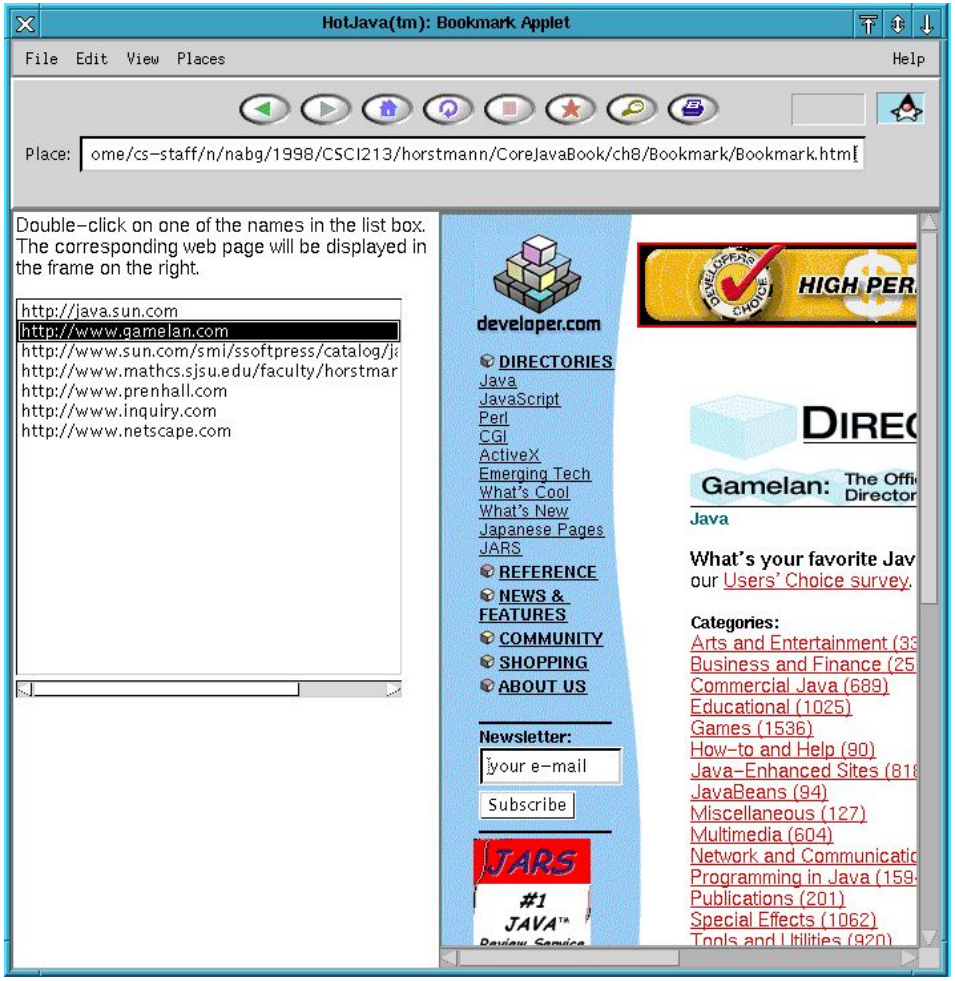
```
public void paint(Graphics g){
    int i;
     int n = Integer.parseInt(getParameter("values").trim());
     double minValue = 0;
     double maxValue = 0;
     for (i = 0; i < values.length; i++)
     {  if (minValue > values[i]) minValue = values[i];
        if (maxValue < values[i]) maxValue = values[i];
     }

     Dimension d = getSize();
     int clientWidth = d.width;
     int clientHeight = d.height;
     int barWidth = clientWidth / n;
```

```
public void paint(Graphics g){
    ...
    int barWidth = clientWidth / n;
    Font titleFont = new Font("Helvetica", Font.BOLD, 20);
    FontMetrics titleFontMetrics  = g.getFontMetrics(titleFont);
    Font labelFont = new Font("Helvetica", Font.PLAIN, 10);
    FontMetrics labelFontMetrics   = g.getFontMetrics(labelFont);
    int titleWidth = titleFontMetrics.stringWidth(title);
    int y = titleFontMetrics.getAscent();
    int x = (clientWidth - titleWidth) / 2;
    g.setFont(titleFont); g.drawString(title, x, y);
    int top = titleFontMetrics.getHeight();
    int bottom = labelFontMetrics.getHeight();
```

```
public void paint(Graphics g){
    ...
   int bottom = labelFontMetrics.getHeight();
   if (maxValue == minValue) return;
   double scale = (clientHeight - top - bottom)
        / (maxValue - minValue);
   y = clientHeight - labelFontMetrics.getDescent();
   g.setFont(labelFont);
   for (i = 0; i < n; i++) {   ... }
  }
```

```
public void paint(Graphics g){
    ...
  for (i = 0; i < n; i++)
  {  int x1 = i * barWidth + 1; int y1 = top;
     int height = (int)(values[i] * scale);
     if (values[i] >= 0)  y1 += (int)((maxValue - values[i]) * scale);
       else  {  y1 += (int)(maxValue * scale); height = -height; }
     g.setColor(Color.red);  g.fillRect(x1, y1, barWidth - 2, height);
     g.setColor(Color.black); g.drawRect(x1, y1, barWidth - 2, height);
     int labelWidth
         = labelFontMetrics.stringWidth(names[i]);
     x = i * barWidth + (barWidth - labelWidth) / 2;
     g.drawString(names[i], x, y);
    }
  }
```


HotJava(tm): Bookmark Applet
File Edit View Places Help
Place: ome/cs-staff/n/nabg/1998/CSCI213/horstmann/CoreJavaBook/ch8/Bookmark/Bookmark.html
Double-click on one of the names in the list box.
The corresponding web page will be displayed in
the frame on the right.
http://java.sun.com
http://www.gamelan.com
http://www.sun.com/smi/ssoftpress/catalog/j
http://www.mathcs.sjsu.edu/faculty/horstmar
http://www.prenhall.com
http://www.inquiry.com
http://www.netscape.com
developer.com
DIRECTORIES
Java
JavaScript
Perl
CGI
ActiveX
Emerging Tech
What's Cool
What's New
Japanese Pages
JARS
REFERENCE
NEWS & FEATURES
COMMUNITY
SHOPPING
ABOUT US
Newsletter:
your e-mail
Subscribe
JARS
#1
JAVA™
HIGH PER
DIREC
Gamelan: The Offi Director
Java
What's your favorite Jav
our Users' Choice survey.
Categories:
Arts and Entertainment (33
Business and Finance (25
Commercial Java (689)
Educational (1025)
Games (1536)
How-to and Help (90)
Java-Enhanced Sites (81
JavaBeans (94)
Miscellaneous (127)
Multimedia (604)
Network and Communicatio
Programming in Java (159
Publications (201)
Special Effects (1062)
Tools and Utilities (920)

Horstmann's bookmark applet

- Illustrates communication with AppletContext
  - request that AppletContext display a different page (display is in frame, could have used a separate window)
- Multiple HTML files
  - frameset using two frames
  - contents for frames

```
<HTML>  <HEAD>
<TITLE>Bookmark Applet</TITLE>
</HEAD>
<FRAMESET COLS="320,*">
<FRAME NAME="left" SRC="Left.html" MARGINHEIGHT=2
    MARGINWIDTH=2
  SCROLLING = "no" NORESIZE>
<FRAME NAME="right" SRC="Right.html" MARGINHEIGHT=2
    MARGINWIDTH=2
  SCROLLING = "yes" NORESIZE>
</FRAMESET>
</HTML>
```

Bookmark.java (HTML file that defines "frameset" with contents Left.html and Right.html)

```
<HTML>
<TITLE>A Bookmark Applet</TITLE>
<BODY>
Double-click on one of the names in the list box. The corresponding web page
will be displayed in the frame on the right.
<P>
<APPLET CODE="Bookmark.class" WIDTH=290 HEIGHT=300>
<PARAM NAME=link_1 VALUE="http://java.sun.com">
...
<PARAM NAME=link_6 VALUE="http://www.inquiry.com">
<PARAM NAME=link_7 VALUE="http://www.netscape.com">
</APPLET>
</BODY>
</HTML>
```

Left.html (HTML file that contains <applet ... > and associated <param ...>s)

```
<HTML>
<TITLE>
Web pages will be displayed here.
</TITLE>
<BODY>
Double-click on one of the names in the list box to the left. The web page
will be displayed here.
</BODY>
</HTML>
```

Right.html (HTML file that contains initial filler for right hand frame)

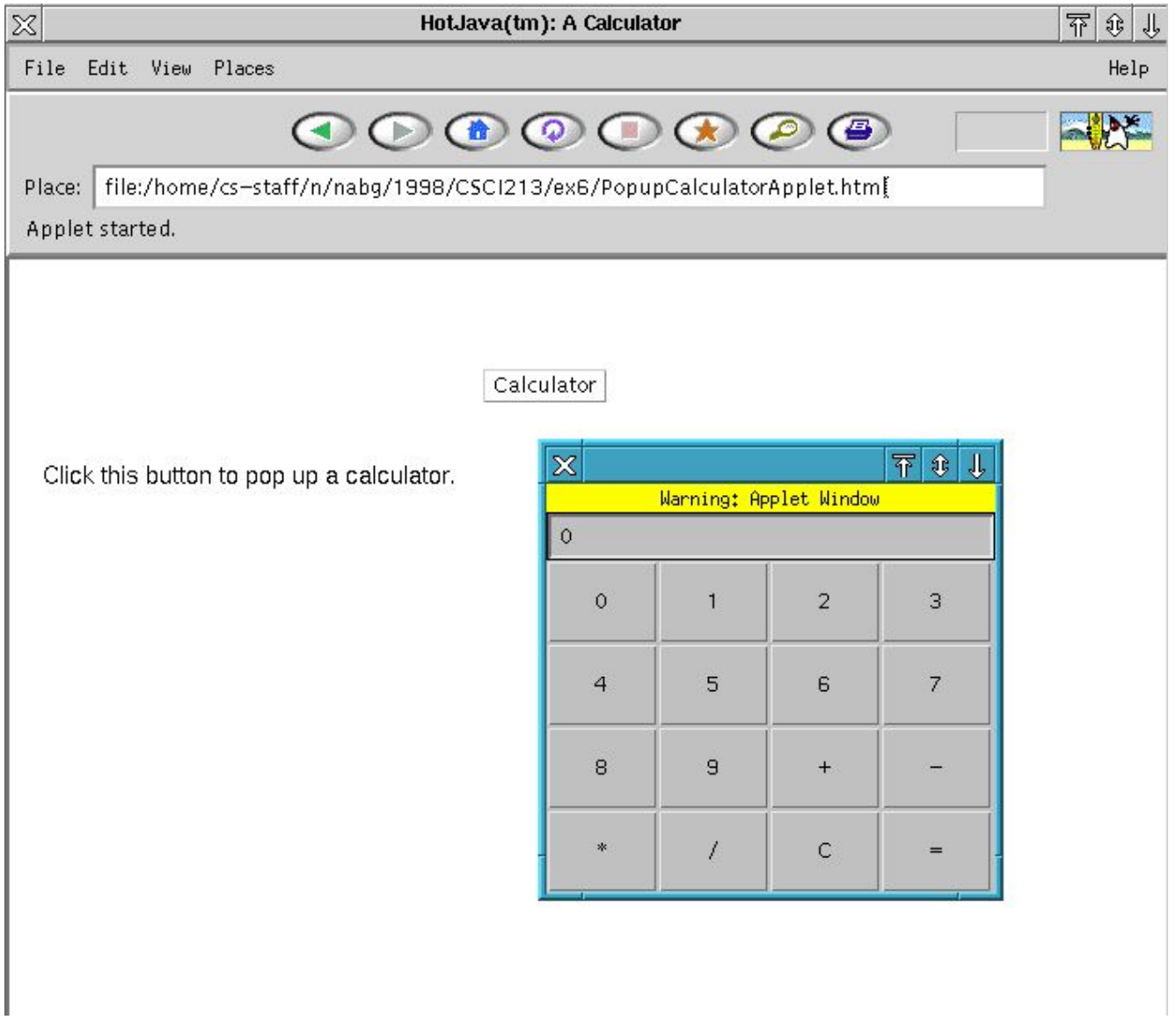
Bookmark.java

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import java.net.*;
import java.io.*;
public class Bookmark extends Applet implements ActionListener{
  public void init() {  ... }
  public void actionPerformed(ActionEvent evt)   {    ... }

  private List links = new List(10, false);
}
```

- A List GUI element can respond to a double click by reporting an ActionEvent
  - something has to handle that event
  - might as well be the Applet, no need for any other object

```
public void init(){
    setLayout(new BorderLayout());
    add("Center", links);
    links.addActionListener(this);
    int i = 1;
    String s;
    while ((s = getParameter("link_" + i)) != null) {  links.add(s);  i++; }
}
public void actionPerformed(ActionEvent evt) {
    String arg = evt.getActionCommand();
     try
      {  AppletContext context = getAppletContext();
        URL u = new URL((String)arg);
        context.showDocument(u, "right");
      } catch(Exception e) {  showStatus("Error " + e); }
}
```


HotJava(tm): A Calculator
File Edit View Places Help
Place: file:/home/cs-staff/n/nabg/1998/CSCI213/ex6/PopupCalculatorApplet.html
Applet started.
Calculator
Click this button to pop up a calculator.
Warning: Applet Window
0
0 1 2 3
4 5 6 7
8 9 + -
* / C =

PopupCalculator

- Slightly reworked version of Horstmann's example (which didn't work properly when I tried it)
- Main new feature:
  - an Applet can open additional windows
  - note "security feature" - such windows should be clearly differentiated from standard browser windows

HTML for calculator

```
<HTML>
<TITLE>A Calculator</TITLE>
<BODY>
Click this button to pop up a calculator.
<APPLET ALIGN=MIDDLE
    CODE="PopupCalculatorApplet.class" WIDTH=100
    HEIGHT=150>
</APPLET>
</BODY>
</HTML>
```

Applet & Calculator classes

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

class Calculator extends Frame implements
    ActionListener, WindowListener { ... }
public class PopupCalculatorApplet extends Applet
    implements ActionListener {
    ...
   private Frame calc = new Calculator();
}
```

```
public class PopupCalculatorApplet extends Applet
    implements ActionListener {
   public void init() {
     Button calcButton = new Button("Calculator");
     calcButton.addActionListener(this);
     add(calcButton);
     calc.setVisible(false);
     }
   public void actionPerformed(ActionEvent evt) {
     if (calc.isVisible()) calc.setVisible(false);
     else calc.show();
     }
   private Frame calc = new Calculator();
}
```

```
class Calculator extends Frame implements ActionListener,
    WindowListener {
public void windowClosing(WindowEvent e) { setVisible(false); }
...
public void windowDeactivated(WindowEvent e) { }

   public Calculator() {  ...  }
   private void addButton(Container c, String s) {      ...  }
   public void actionPerformed(ActionEvent evt) {       ...  }
   public void calculate(int n) {      ...  }
   private TextField display;
   private int arg = 0;
   private String op = "=";
   private boolean start = true;
}
```

```
public Calculator() {
    display = new TextField("0");  display.setEditable(false);
    add(display, "North");
    Panel p = new Panel();p.setLayout(new GridLayout(4,4));
    for(int i=0;i<=9;i++)
     addButton(p, ""+(char)('0'+i));
    addButton(p, "+");  addButton(p, "-");
    addButton(p, "*");  addButton(p, "/");
    addButton(p, "C"); addButton(p, "=");
    add(p, "Center");
   this.addWindowListener(this);
  }
   private void addButton(Container c, String s) {
    Button b = new Button(s);   c.add(b);
    b.addActionListener(this); }
```

```
public void actionPerformed(ActionEvent evt) {
    String s = evt.getActionCommand();   char ch = s.charAt(0);
    if ('0' <= ch && ch <= '9' ) {
     if (start) display.setText(s);
          else display.setText(display.getText() + s);
          start = false;  }
     else {
     if (start) {
          if (s.equals("-"))  { display.setText(s); start = false; }
          else if (s.equals("C")) { arg = 0; display.setText("" + arg); }
               else op = s;
               }
          else {   calculate(Integer.parseInt(display.getText()));
               op = s; start = true;
               }
          }
  }
```

```
public void calculate(int n) {
    if (op.equals("+")) arg += n;
    else if (op.equals("-")) arg -= n;
    else if (op.equals("*")) arg *= n;
    else if (op.equals("/")) arg /= n;
    else if (op.equals("=")) arg = n;
    display.setText("" + arg);
  }
```