

Building Web Applications with Servlets and JavaServer Pages

David Janzen

Assistant Professor of Computer Science

Bethel College

North Newton, KS

<http://www.bethelks.edu/djanzen>

djanzen@bethelks.edu



*Bethel
College*

KANSAS

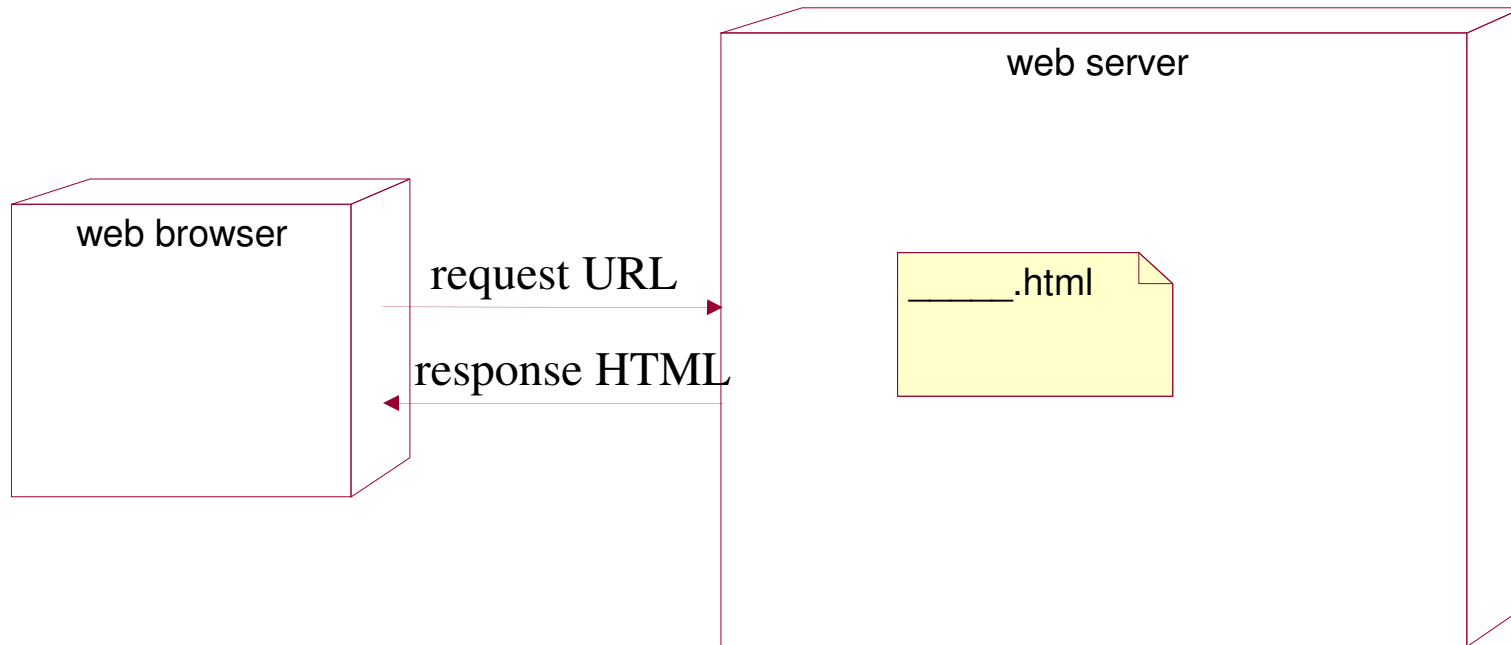
Acknowledgments

- UML is a trademark of Object Management Group, Inc. in the U.S. and other countries
- Rational software and courseware used with permission by Rational Software Corp. and through Rational's SEED (Software Engineering for Educational Development) program
 - http://www.rational.com/corpinfo/college_relations/seed/

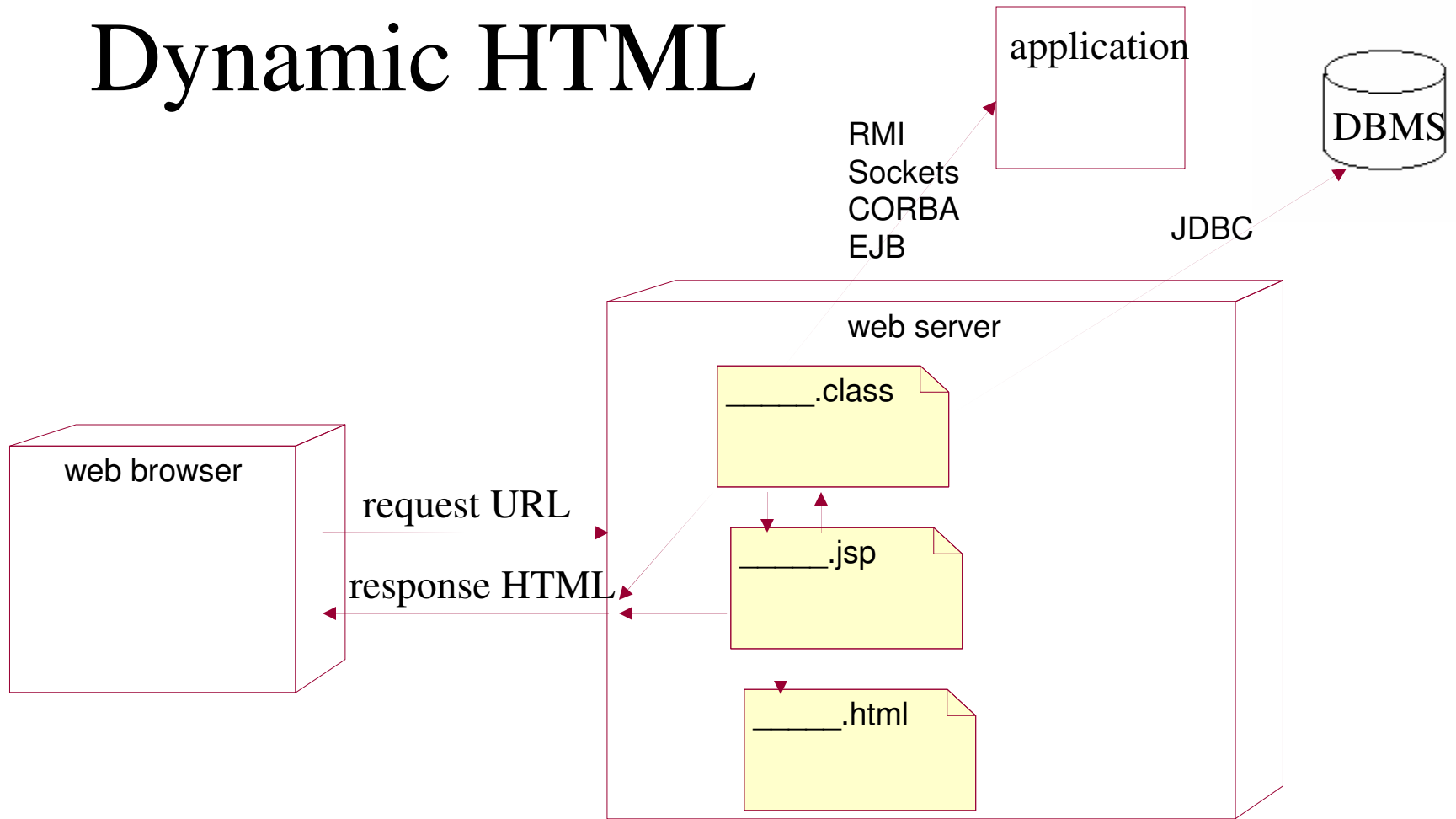
Web Development Overview

- static html
 - text file containing html tags created manually
 - may include some client-side scripts (e.g. JavaScript)
- dynamic html
 - html file produced at time of request
 - cgi, php, asp, jsp, Servlets
- active html
 - html contains a program that runs at the client inside a web browser
 - Java applets

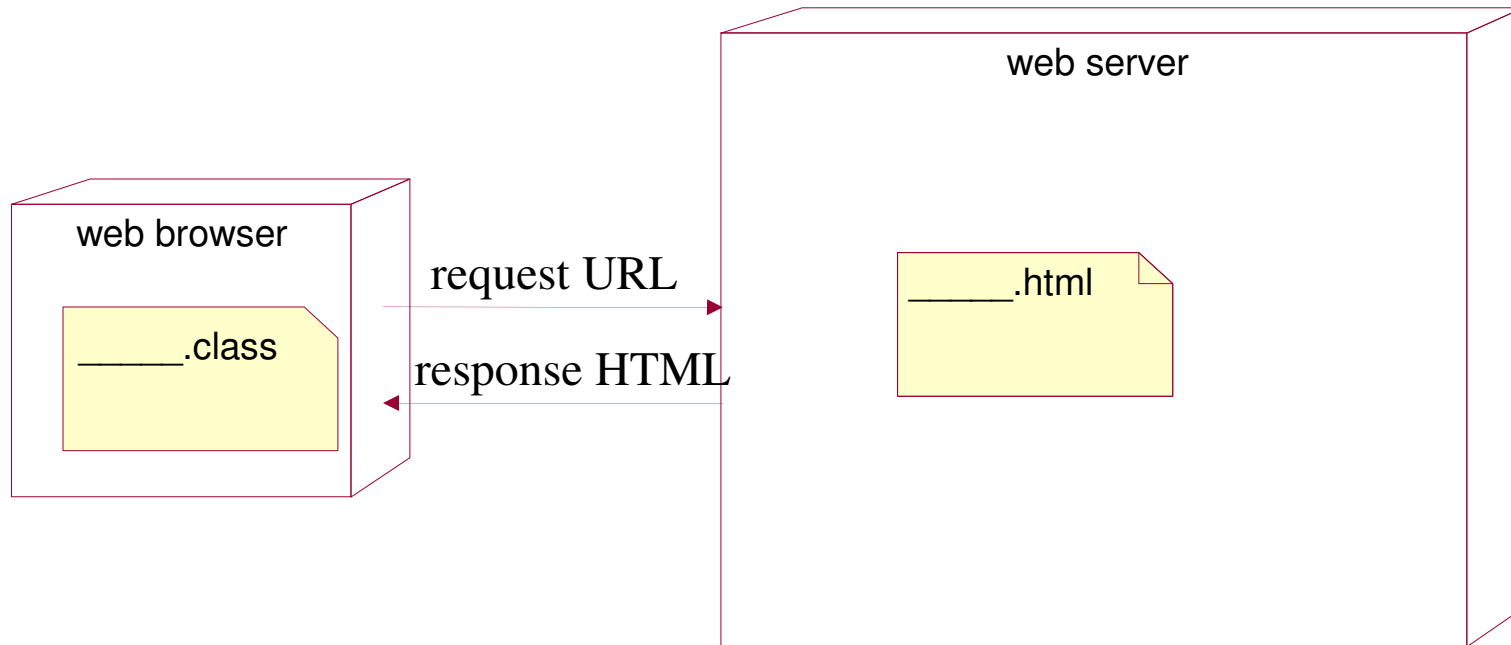
Static HTML



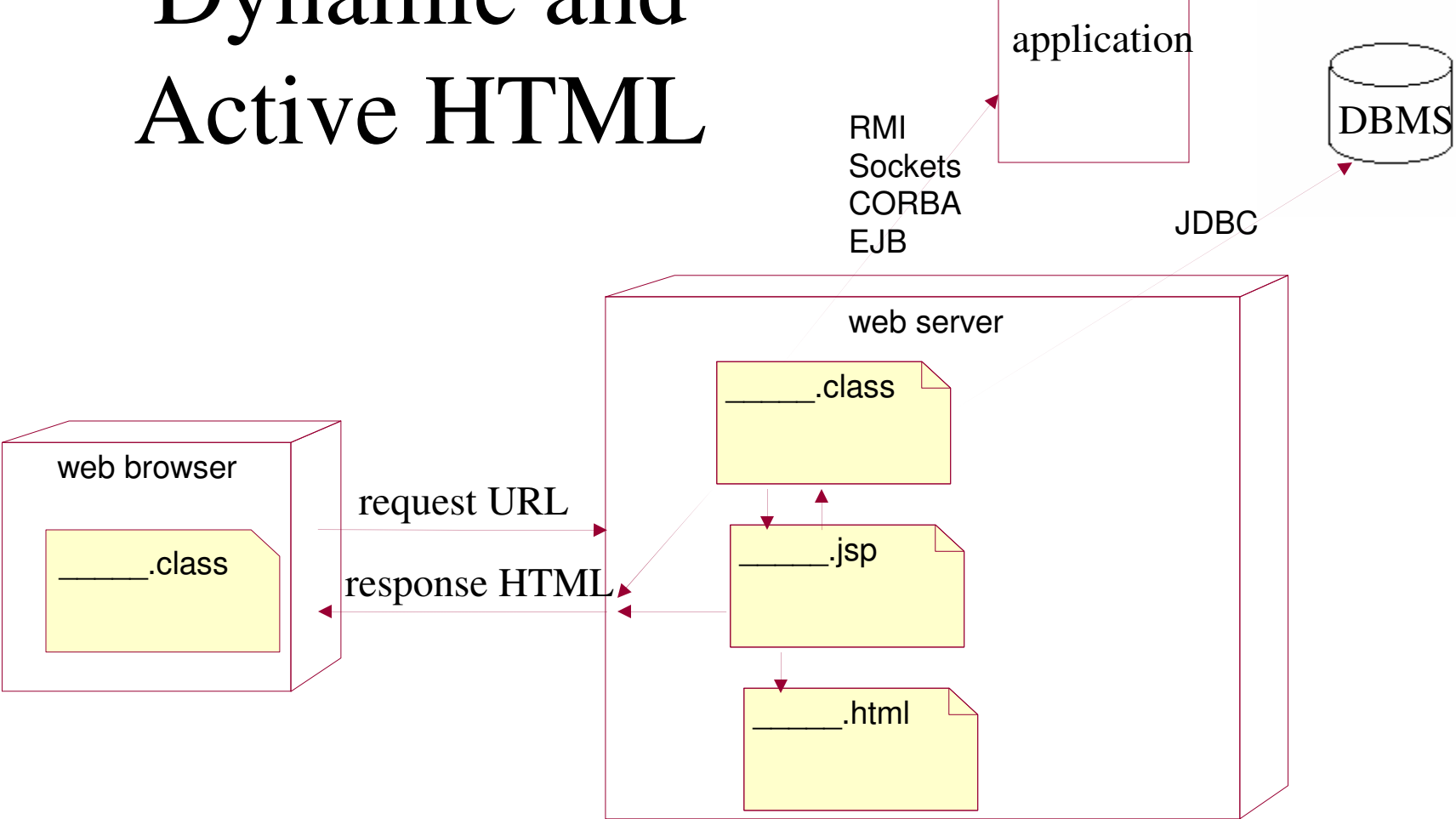
Dynamic HTML



Active HTML



Dynamic and Active HTML

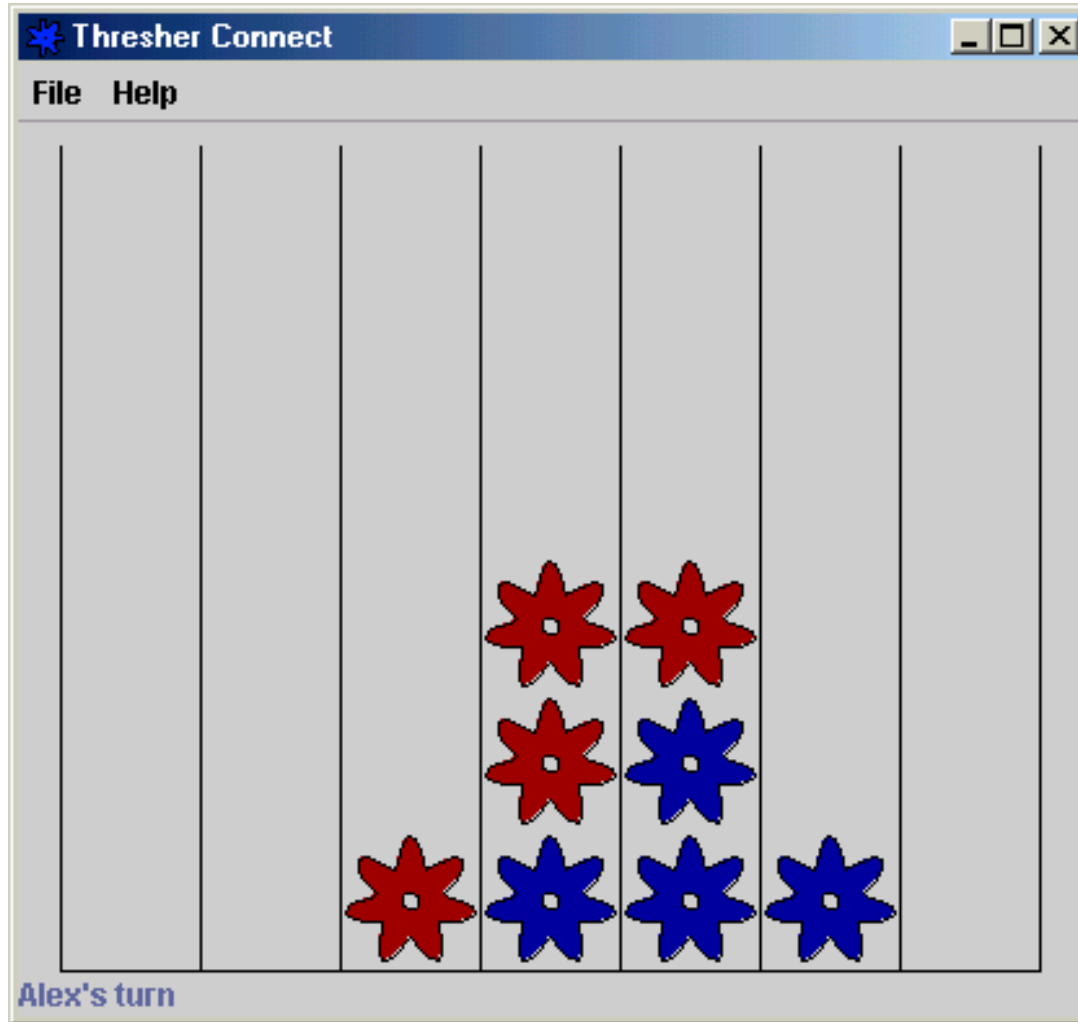


Java Confusion

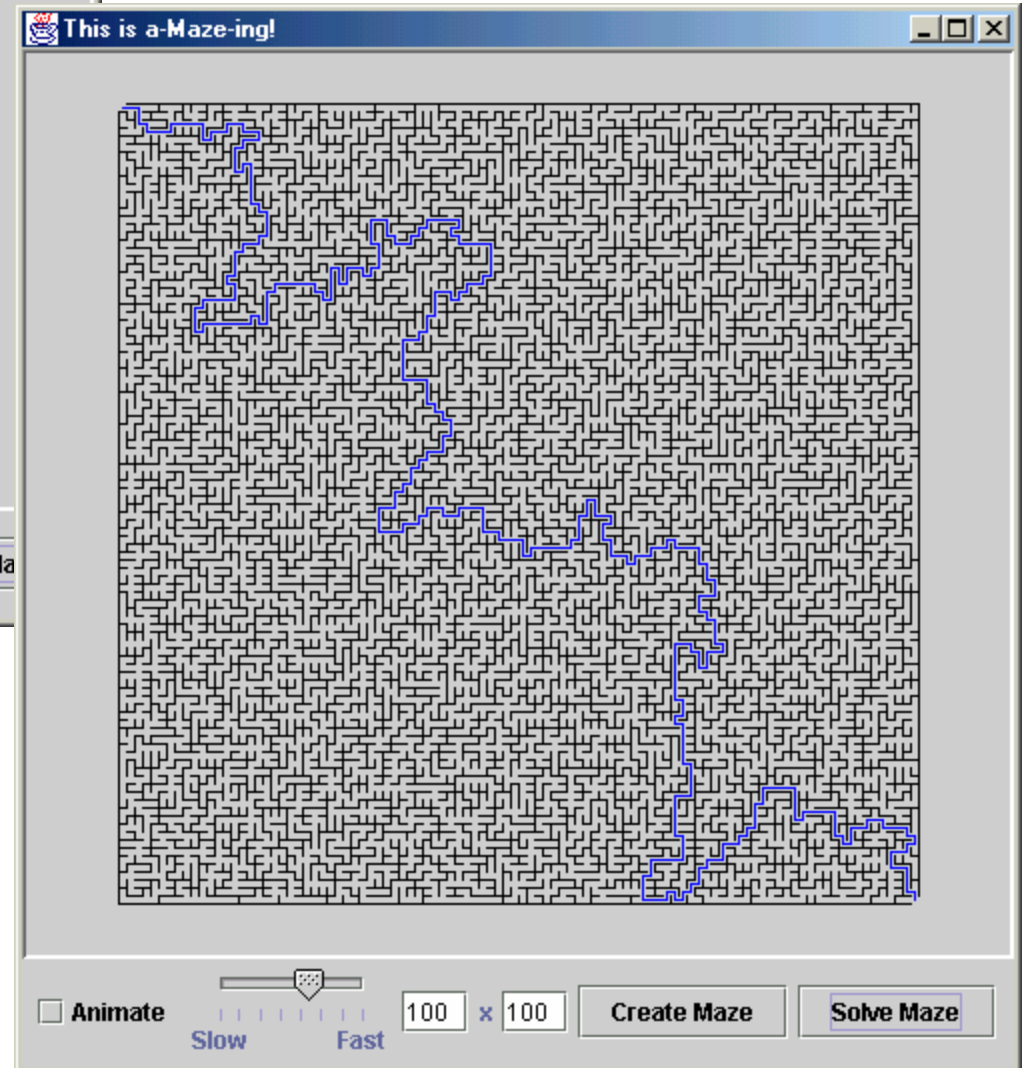
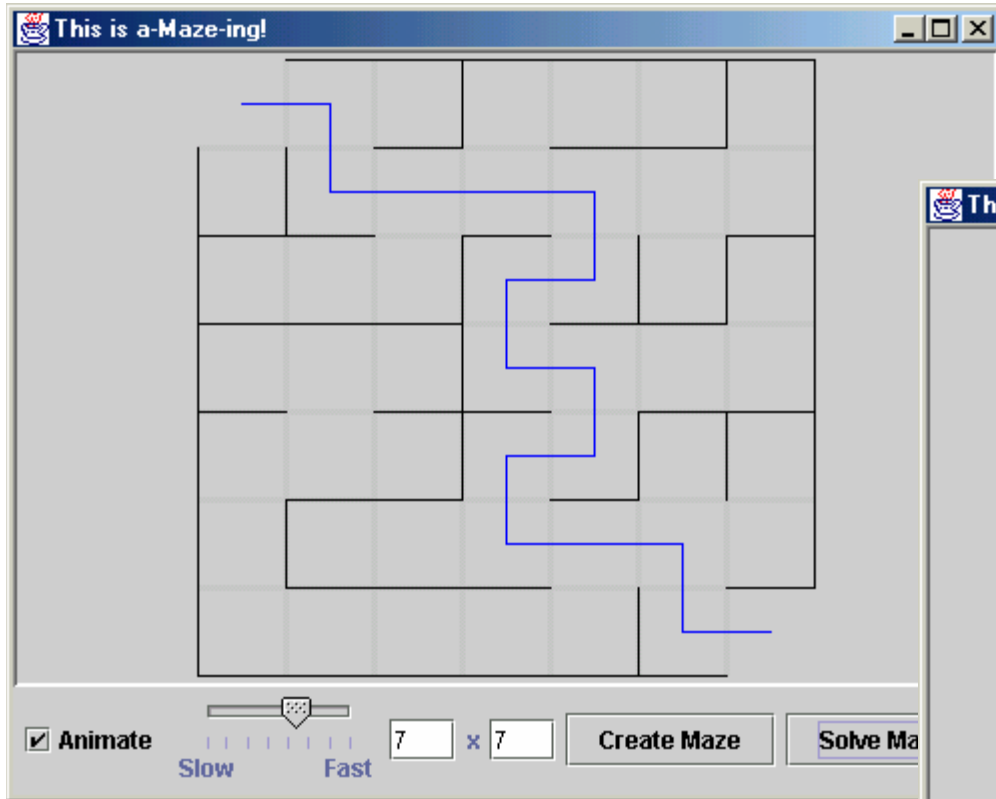
- Java Applications*
 - stand-alone executable applications
- Java Applets*
 - applications that run within a web-browser
- Java Servlets
 - applications that run within a web-server
- JavaScript
 - scripts that run within a web-browser
 - not really Java at all

* available with original 1995 Java release

Java Application



Java Application



Java Applet

Animated AVL Tree - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail News

Address <http://www1.ics.uci.edu/~goodrich/dsa/trees/avltree.html> Go Links >>

Insert Delete Search 32

```
graph TD; 27[27] --- 15[15]; 27 --- 29[29]; 29 --- 32[32];
```

New Node : 32 Added as Right Child of Node : 29

Done Internet

Java Applet

The screenshot shows a Microsoft Internet Explorer browser window titled "Tic Tac Toe Applet with Modified Minimax Code - Microsoft Internet Explorer". The address bar shows the file path: "D:\Bethel\ccsc\James\Tic_Tac_Toe_Applet_with_Modified_Minimax_Code.html".

On the left side of the browser window, there is a button labeled "TicTacToe". Below it, the source code of the HTML file is displayed in Notepad. The code includes a Java applet tag with the following attributes: `<APPLET CODE="SampleApplet.class" WIDTH=120 HEIGHT=60>` and an error message: `Error! You must use a Java-enabled browser.`.

In the center of the browser window, a Java applet window titled "Tic Tac Toe" is displayed. The applet shows a 3x3 Tic Tac Toe board. The board contains a blue 'X' in the top-left cell, a red 'O' in the top-right cell, and a red 'O' in the middle-left cell. The rest of the board is empty.

```
public MoveInfo findXMove(int alpha, int beta,
{
    treeNodes++;
    if(depth > treeDepth)
        treeDepth = depth;

    int i, responseValue;
    int value, bestMove = 0;
    MoveInfo quickWinInfo;

    if( fullBoard() )
        value = DRAW;
    else if( (quickWinInfo = immediateWin(PLA
        return quickWinInfo;
    else
    {
        if(pruning)
            value = alpha;
        else
            value = PLAYERO;

        if(depth == 0 && (quickWinInfo = immediateWin(PLAYERO)) )
            blockMove = quickWinInfo.move;
```

JavaScript

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page titled "Simple JavaScript Button". The page contains a button labeled "Don't Click Me". A small dialog box is open over the button, displaying a warning icon and the message "I told you not to click!".

The Notepad window shows the source code for the page:

```
File Edit Search Help
<!-- Taken from Core Web Programming from -->
<!-- Prentice Hall and Sun Microsystems Press, -->
<!-- http://www.corewebprogramming.com/. -->
<!-- &copy; 2001 Marty Hall and Larry Brown; -->
<!-- may be freely used or adapted. -->

<HTML>
<HEAD>
  <TITLE>Simple JavaScript Button</TITLE>

<SCRIPT TYPE="text/javascript">
<!--
function dontClick() {
  alert("I told you not to click!");
}
// -->
</SCRIPT>
</HEAD>

<BODY BGCOLOR="WHITE">
<H1>Simple JavaScript Button</H1>

<FORM>
  <INPUT TYPE="BUTTON"
    VALUE="Don't Click Me"
    onClick="dontClick()">
</FORM>

</BODY>
</HTML>
```

The browser's address bar shows the URL: <http://archive.corewebprogramming.com/>. The left sidebar of the browser contains a list of chapters:

- Chapter 15: [Advanced Swing](#)
- Chapter 16: [Concurrent Programming with Java Threads](#)
- Chapter 17: [Network Programming](#)
- Chapter 18: [HTML FORMS](#)
- Chapter 19: [Servlets](#)
- Chapter 20: [JSP](#)
- Chapter 21: [Using Applets As Front Ends To Server-Side Programs](#)
- Chapter 22: [JDBC](#)
- Chapter 23: [XML Processing with Java](#)
- Chapter 24: [JavaScript: Adding Dynamic Content to Web Pages](#)
- Chapter 25:

Introduction to Servlets

- Servlets are Java programs that run inside a web server.
- Servlets allow web servers to receive requests from clients (normally entered in a form on a web page).
- Servlets can perform server-side processing such as interacting with a database or another application
- Servlets can generate dynamic html based on server-side processing and return this to the client.

Servlet Howto

- create an HTML file that invokes a servlet (usually through the FORM ACTION=...)
- create a Java program that does the following:
 - import javax.servlet.*;
 - import javax.servlet.http.*;
 - inherit from HttpServlet
 - override the doGet and doPost methods
 - write the response HTML file using a java.io.PrintWriter to the HTTP response

Introduction to Servlets

- A Simple Example
 - collect username and password
 - reply with welcome page
 - or Invalid Login if password is not “verysecret”


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>CCSC Tutorial Demo</TITLE>
  </HEAD>
  <BODY>

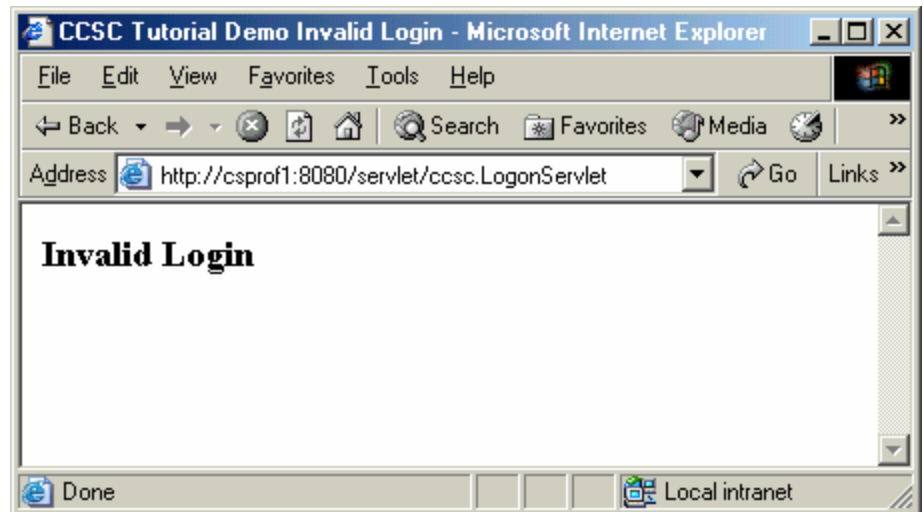
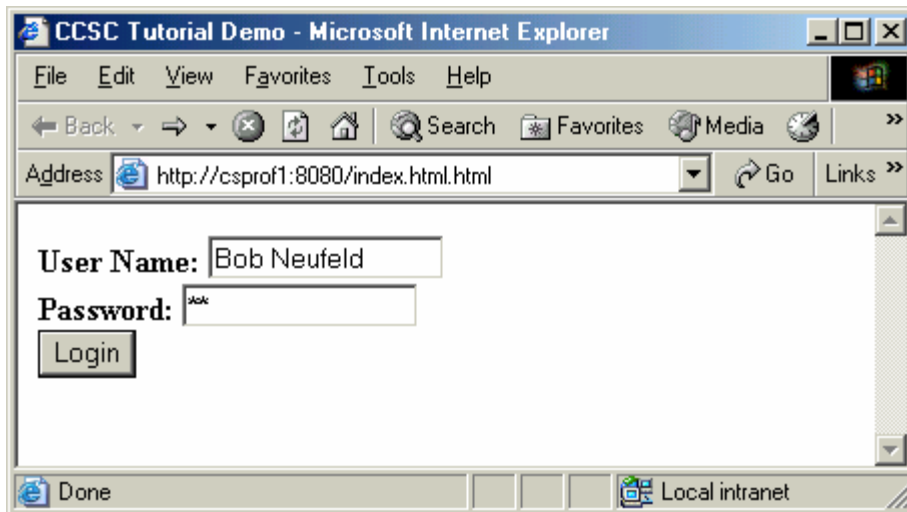
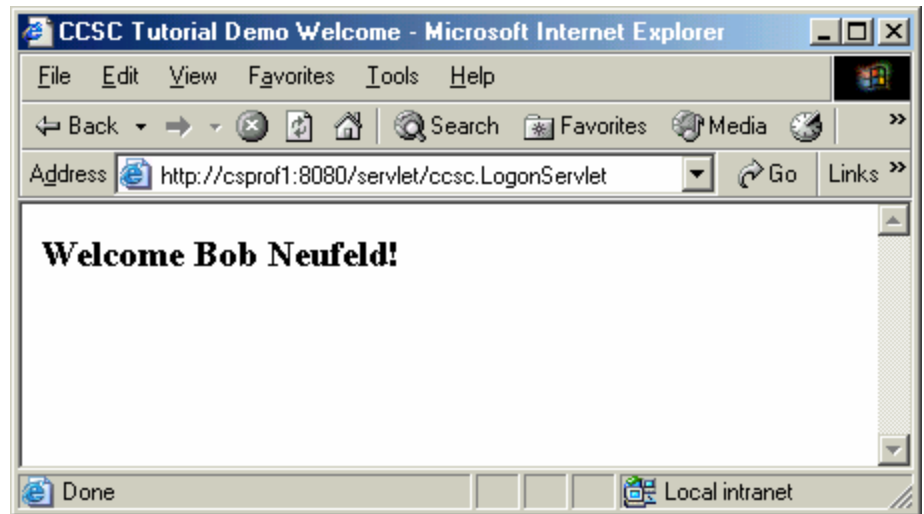
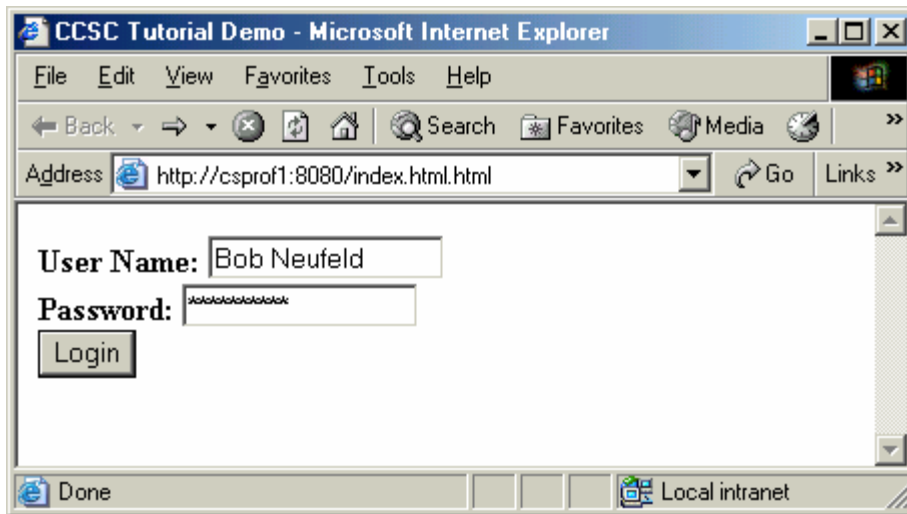
    <FORM NAME="addUser" METHOD="POST"
      ACTION = "/servlet/ccsc.LogonServlet">
      <B>User Name: </B>
      <INPUT NAME = "userName" TYPE = "TEXT"
        MAXLENGTH = "25" SIZE = "15"> <br>
      <B>Password: </B>
      <INPUT NAME = "password" TYPE = "password"
        VALUE = "verysecret" MAXLENGTH = "25" SIZE = "15">
      <br>
      <B><INPUT NAME = "login" VALUE = "Login"
        TYPE = "SUBMIT"></B>
    </FORM>

    <script language="JavaScript">
      document.addUser.userName.focus()
    </script>
  </BODY>
</HTML>
```

```

package ccsc;
import javax.servlet.*;
import javax.servlet.http.*;
public class LogonServlet extends HttpServlet {
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, java.io.IOException {
        response.setContentType("text/html");
        java.io.PrintWriter out = response.getWriter();
        String uName = request.getParameter("userName");
        String pWord = request.getParameter("password");
        if(pWord.equals("verysecret")) {
            out.println("<html><head><title>CCSC Tutorial Demo Welcome</title></head>");
            out.println("<body><h3>Welcome " + uName + "!</h3></body></html>");
        }
        else {
            out.println("<html><head><title>CCSC Tutorial Demo Invalid Login</title></head>");
            out.println("<body><H3>Invalid Login</H3></body></html>");
        }
        out.close();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, java.io.IOException {
        processRequest(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, java.io.IOException {
        processRequest(request, response);
    }
}

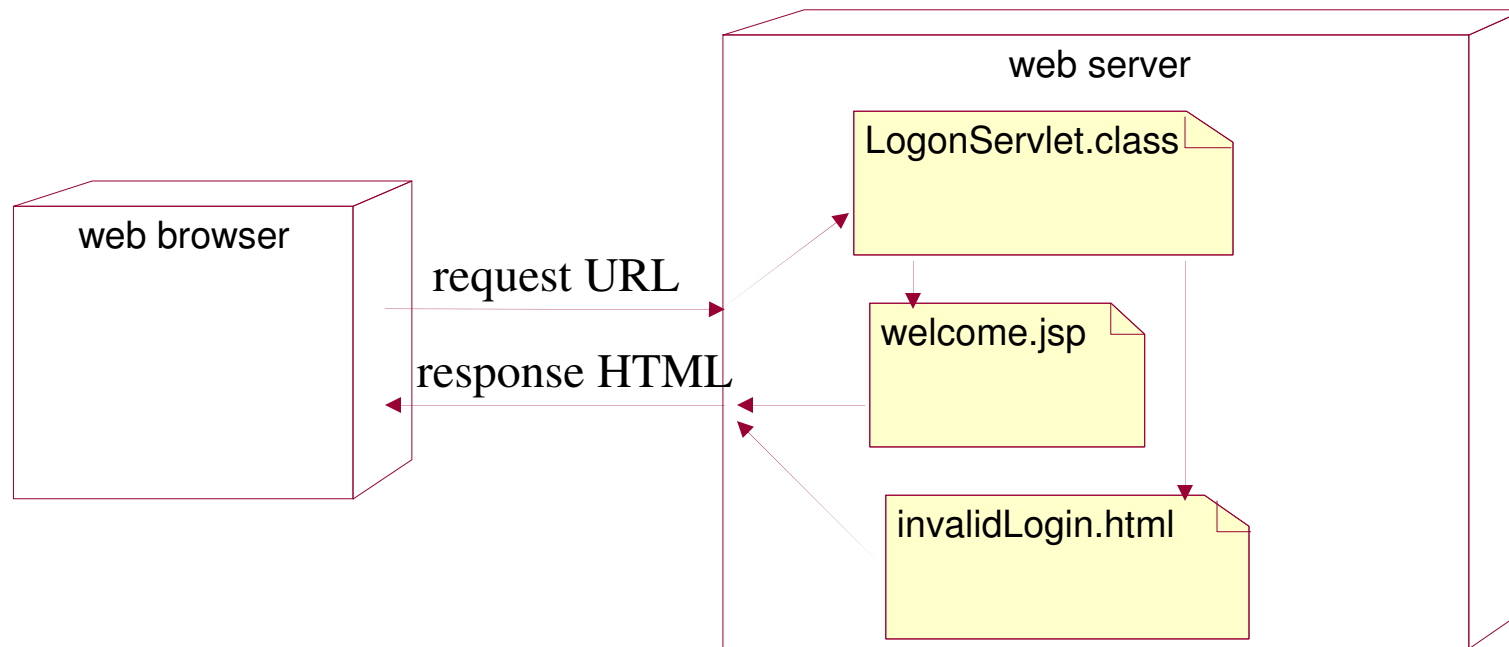
```



JavaServer Pages

- JavaServer Pages provide a layer above Servlets that allow you to mix HTML and calls to Java code in the web server.
- JavaServer Pages are actually converted into Servlets, but JSP's are easier to work with for those familiar with HTML.
- Servlet's usually create Java objects as Java Beans to simplify JSP access.

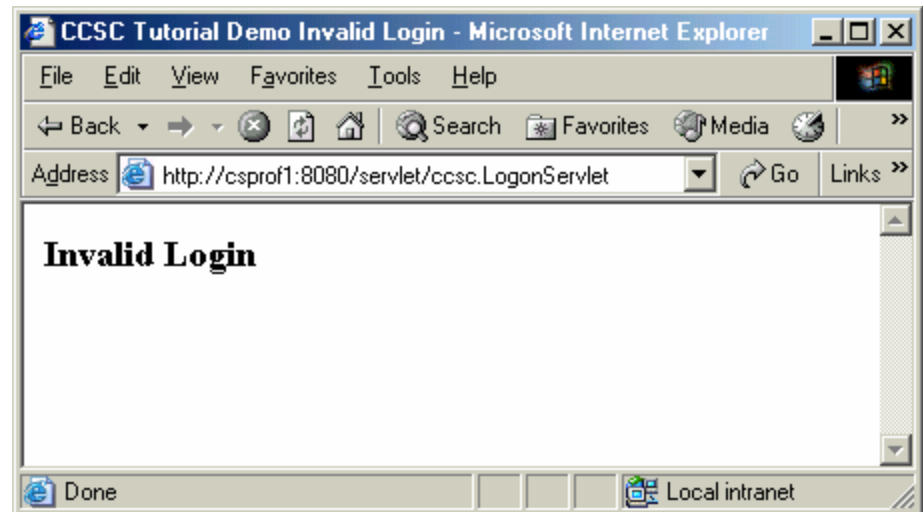
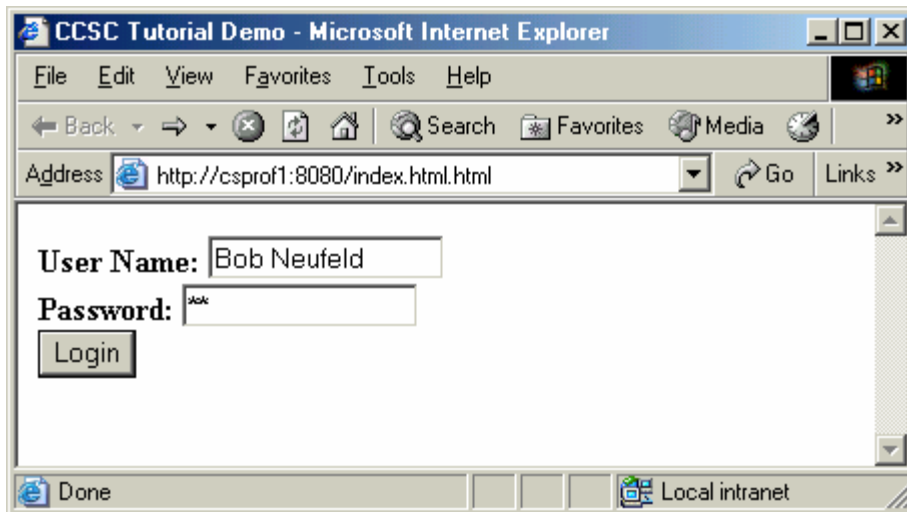
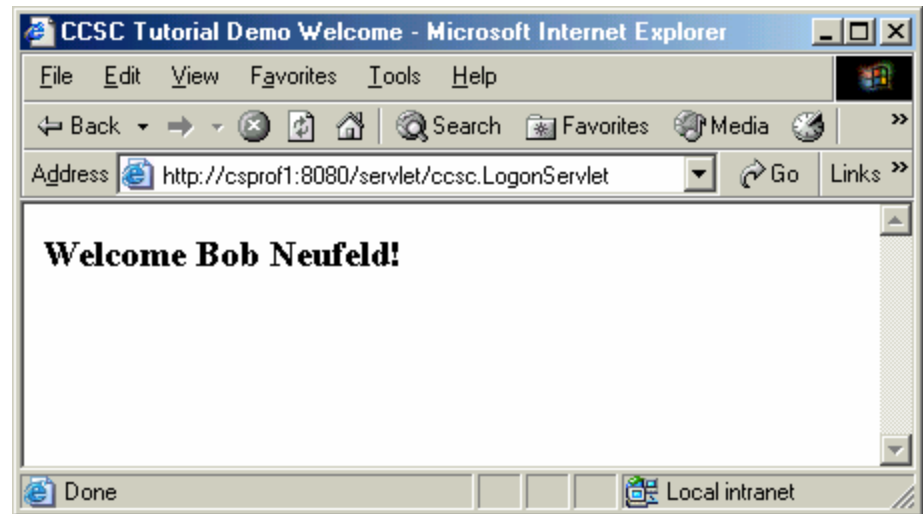
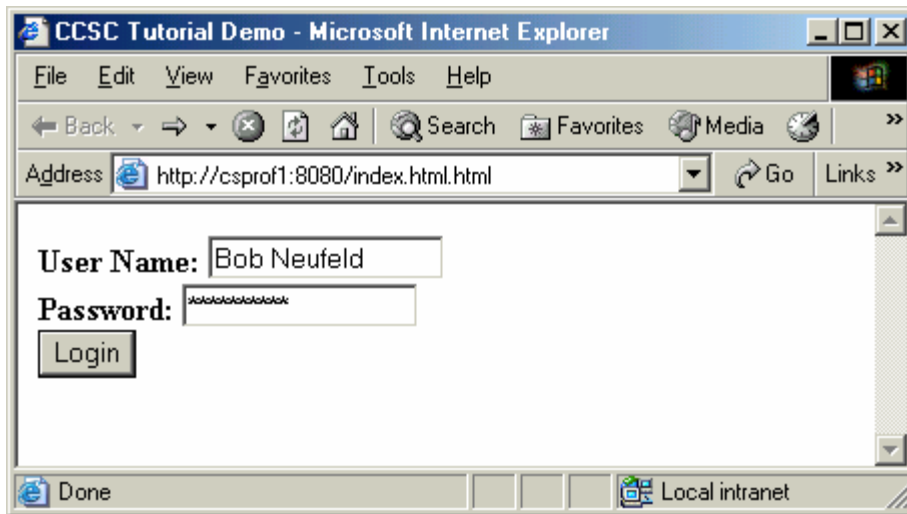
Mixing JavaServer Pages with Servlets



```
package ccsc;
import javax.servlet.*;
import javax.servlet.http.*;
public class LogonServlet extends HttpServlet {
...
    protected void processRequest(HttpServletRequest request,
                                   HttpServletResponse response)
        throws ServletException, java.io.IOException {
        response.setContentType("text/html");
        String uName = request.getParameter("userName");
        String pWord = request.getParameter("password");
        HttpSession session = request.getSession(true);
        if(pWord.equals("verysecret")) {
            session.setAttribute("uName",uName);
            gotoPage("/welcome.jsp",request,response);
        }
        else {
            gotoPage("/invalidLogin.html",request,response);
        }
    }
    private void gotoPage(String address, HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, java.io.IOException {
        RequestDispatcher dispatcher = getServletContext().getRequestDispatcher(address);
        dispatcher.forward(request, response);
    }
...
}
```

welcome.jsp

```
<%@page contentType="text/html"%>  
<html>  
<head><title>CCSC Tutorial Demo Welcome</title></head>  
<body>  
  
    <h3>Welcome <%= session.getAttribute("uName")%>!</h3>  
  
</body>  
</html>
```



JSP Howto

- Create a file with a .jsp extension
- include static HTML
- include JSP Scripting Elements
 - Expressions `<%= expression %>`
 - Scriptlets `<% code %>`
 - Declarations `<%! Code %>`
 - Directives `<%@ variable directive %>`
 - Comments `<%-- JSP Comment -->`
- Use pre-defined variables
 - request `HttpServletRequest`
 - response `HttpServletResponse`
 - session `HttpSession`
 - out `PrintWriter`

JSP Howto

- Include another file

```
<%@ include file="commontop.html" %>
```

- Import from the Java library

```
<%@ page import="java.util.*" %>
```

- Declare and use a variable

```
<%! private int accessCount = 0 %>
```

```
<H2>Accesses to page since server reboot:
```

```
<%= ++accessCount %></H2>
```

- Access session and request information

```
<H2>Current time: <%= new java.util.Date() %></H2>
```

```
<H2>Remote Host: <%= request.getRemoteHost() %></H2>
```

```
<H2>Session ID: <%= session.getID() %></H2>
```

JSP Howto

- Assuming a Servlet has done the following:

```
Vehicle v = new Vehicle();  
v.pic1Filename = "Honda1996.jpg";  
v.year = 1996;  
session.setAttribute("vehDetail",v);
```

- In the JSP, access a Java object in the server

```
<jsp:useBean id="vehDetail" class = "ccsc.Vehicle"  
  scope = "session"/>  
<% out.println("<H3>Year: " + vehDetail.year + "</H3>");  
  out.println("<IMG SRC=/jsp/dirs/vehiclePics/" +  
    vehDetail.pic1Filename +  
    " WIDTH=300 HEIGHT=180>");  
%>
```

Advantages of using Servlets and JavaServer Pages

- Java is widely known
- Java has many applications
 - from GUI's to PDA's to Applets to Servlets
- Java has a rich set of libraries
 - threads
 - networking (sockets, RMI, CORBA)
 - database connectivity through JDBC

Advantages of using Servlets and JavaServer Pages

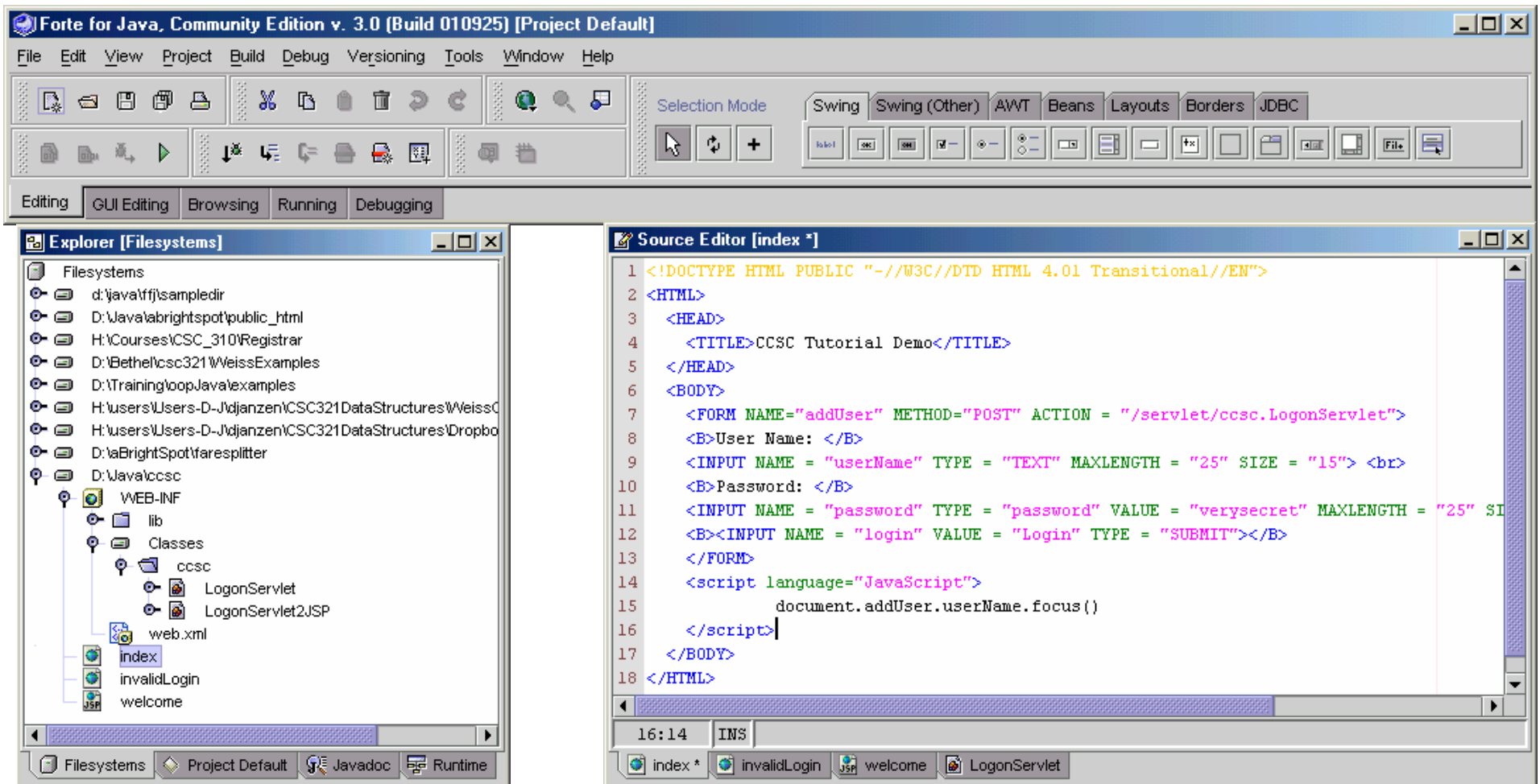
- Java is free
 - <http://java.sun.com> for Java SDK and J2EE
 - <http://jakarta.apache.org> for Tomcat
- Java is portable
 - runs on many OS's and in many Servlet Engines
- Java is efficient
 - multiple concurrent lightweight threads execute a single set of code within the web server
- Java is secure
 - memory restrictions (array out-of-bounds)

Setting up the environment

- Many Servlet engines
 - <http://www.servlets.com/engines/index.html>
- Tomcat on Apache on Linux
- Outsource it through web hosting
 - (3tec.com \$20 + \$7x4months for a semester)
- Forte for Java has Tomcat built-in
 - <http://www.sun.com/forte/ffj/>

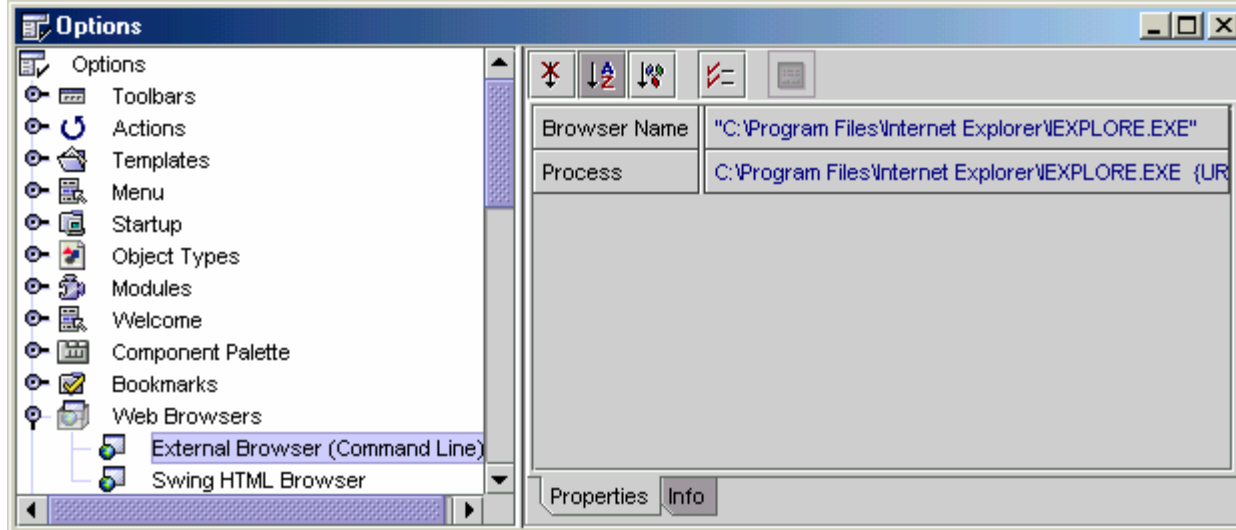
Setting up the environment

- When using Tomcat (stand-alone or in FFJ)
 - place .html and .jsp files in home dir
 - place .class Servlet files in WEB-INF/classes



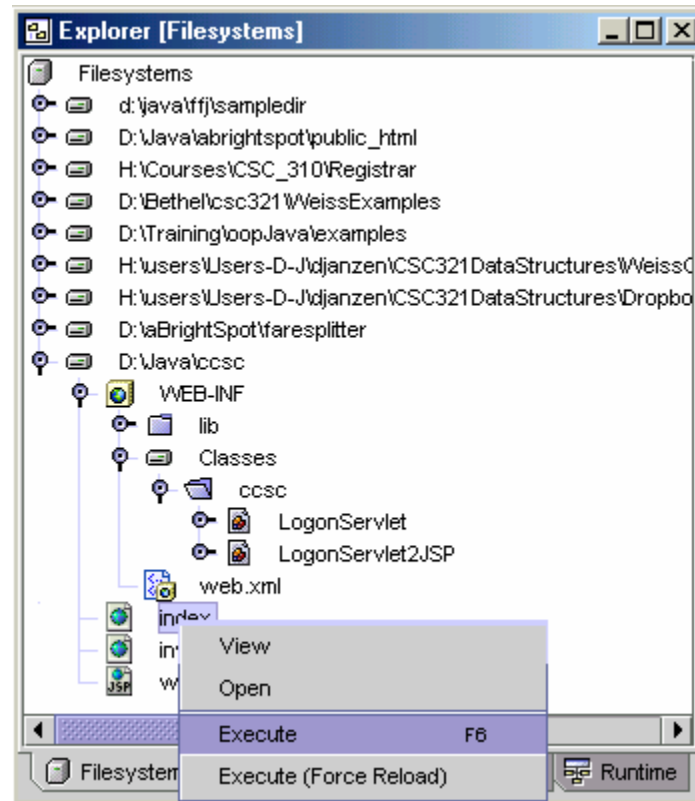
Setting up the environment

- Tips to avoid Pitfalls
 - in Forte for Java
 - add all files through Template Wizard (File->New)
 - set External Browser through Tools->Options



Setting up the environment

- Tips to avoid Pitfalls
 - in Forte for Java
 - Execute web page, don't run it



Sample Web Application

- SeaGypsy Hotel Reservation System
 - Created by Junior/Senior level students
 - CSC361 Software Design and Development
 - January 2001



The Sea Gypsy Motel - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Mail News RSS User

Address <http://dirk.bethelks.edu/seagypsy/roomrequest.html> Go

Links Best of the Web Channel Guide Customize Links Free Hotmail Internet Start Is Your Operating System Genuine >>

Room Request Form

Please Select an Arrival Month and Day

2001 November 5

Please Select a Departure Month and Day


2001 November 8

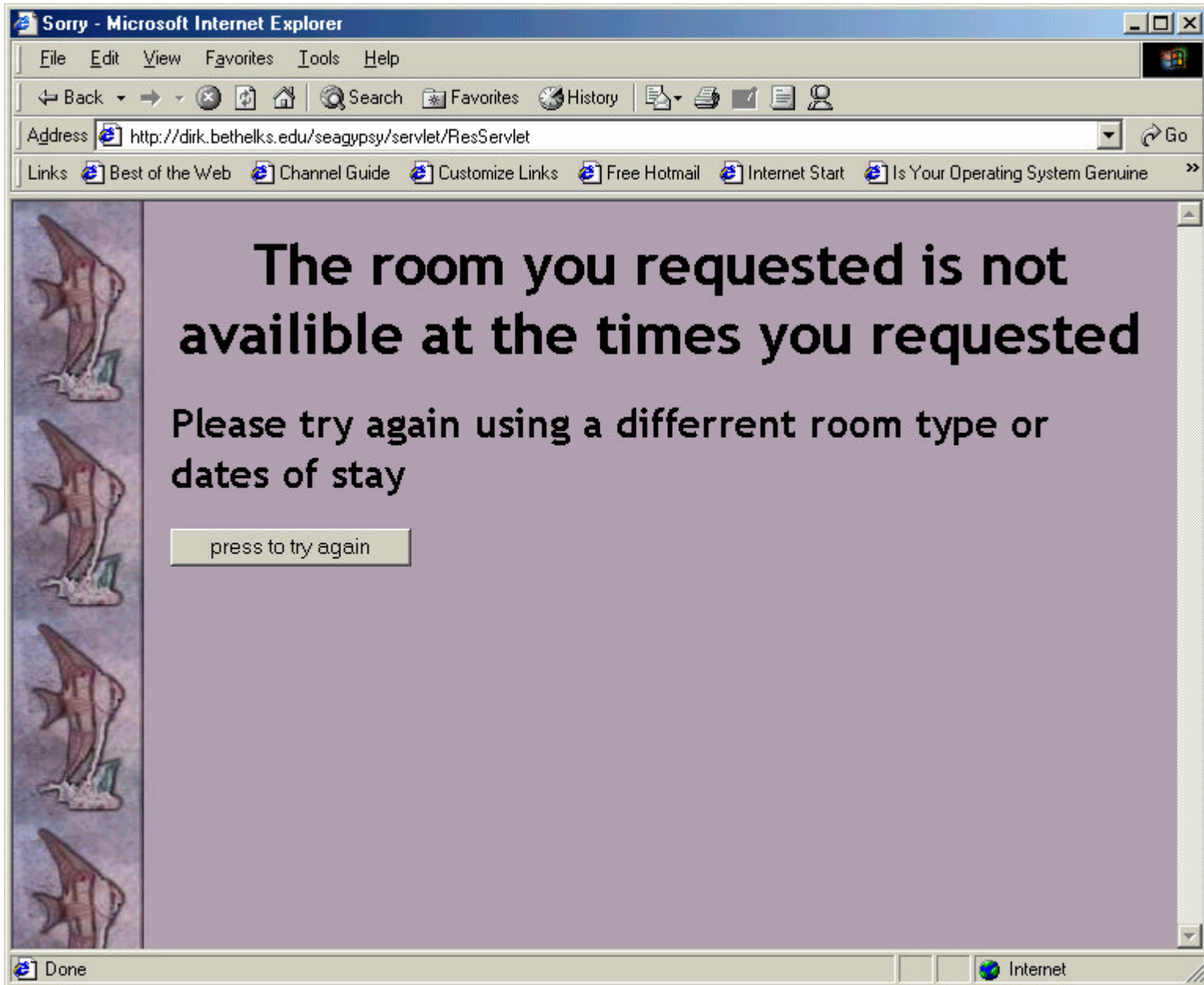
Please select a room type

studio ocean view

Submit Query

Done Internet





The Sea Gypsy Motel - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Mail News RSS User

Address <http://dirk.bethelks.edu/seagypsy/roomrequest.html?redo=press+to+try+again> Go

Links Best of the Web Channel Guide Customize Links Free Hotmail Internet Start Is Your Operating System Genuine >>

Room Request Form

Please Select an Arrival Month and Day

2001 November 20

Please Select a Departure Month and Day


2001 November 26

Please select a room type

studio ocean view

Submit Query

Done Internet



Information Reserve - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Mail News RSS User

Address <http://dirk.bethelks.edu/seagypsy/servlet/ResServlet> Go

Links [Best of the Web](#) [Channel Guide](#) [Customize Links](#) [Free Hotmail](#) [Internet Start](#) [Is Your Operating System Genuine](#) >>

The Sea Gypsy Motel

Room # 102 is available

From 11/ 20/ 2001 to 11/ 26/ 2001

Please fill out the form below and submit it

First Name :

Last Name :

Address :

Phone Number :

Number of people over age 3 :

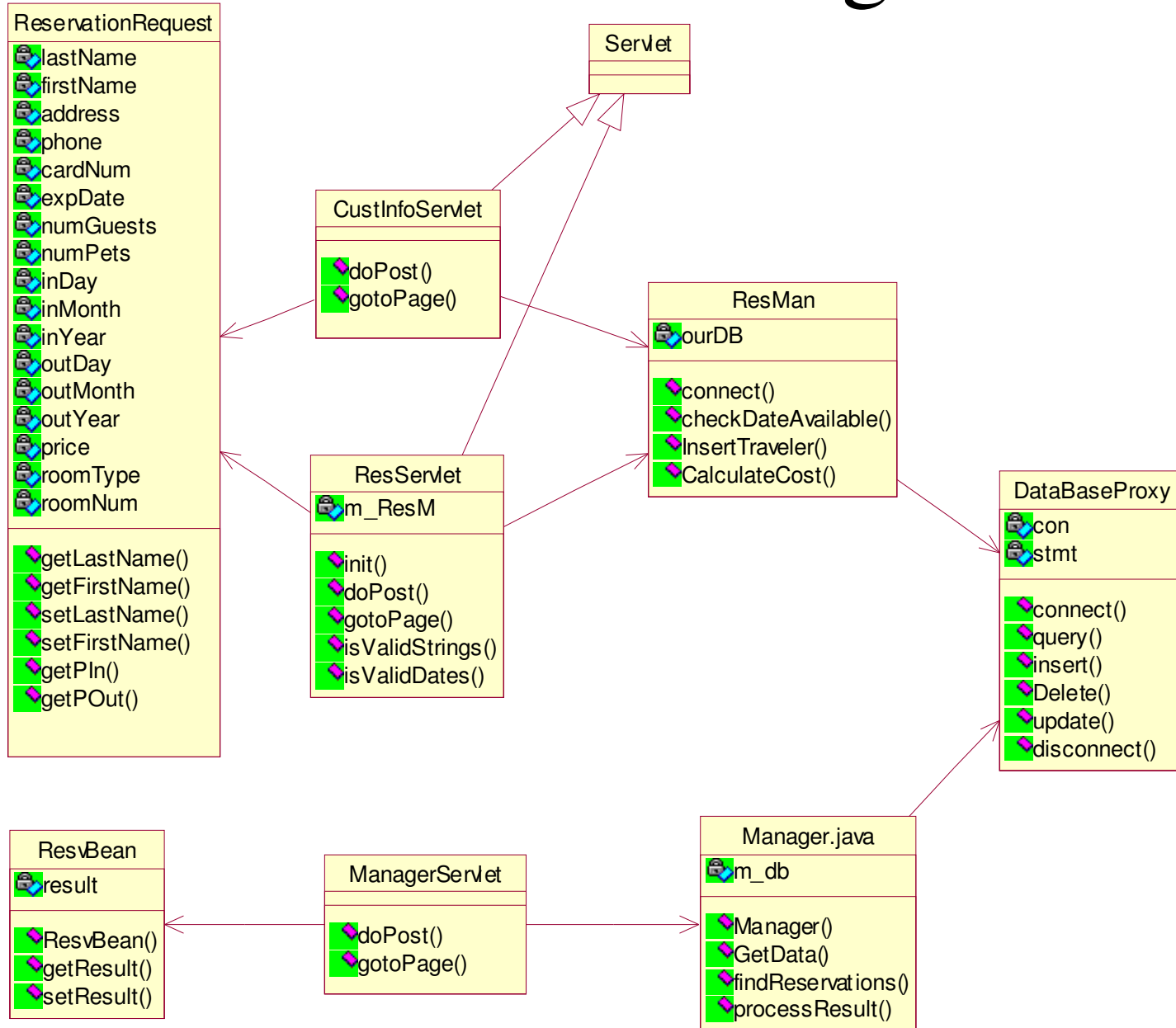
Number of Pets :

Credit Card number :

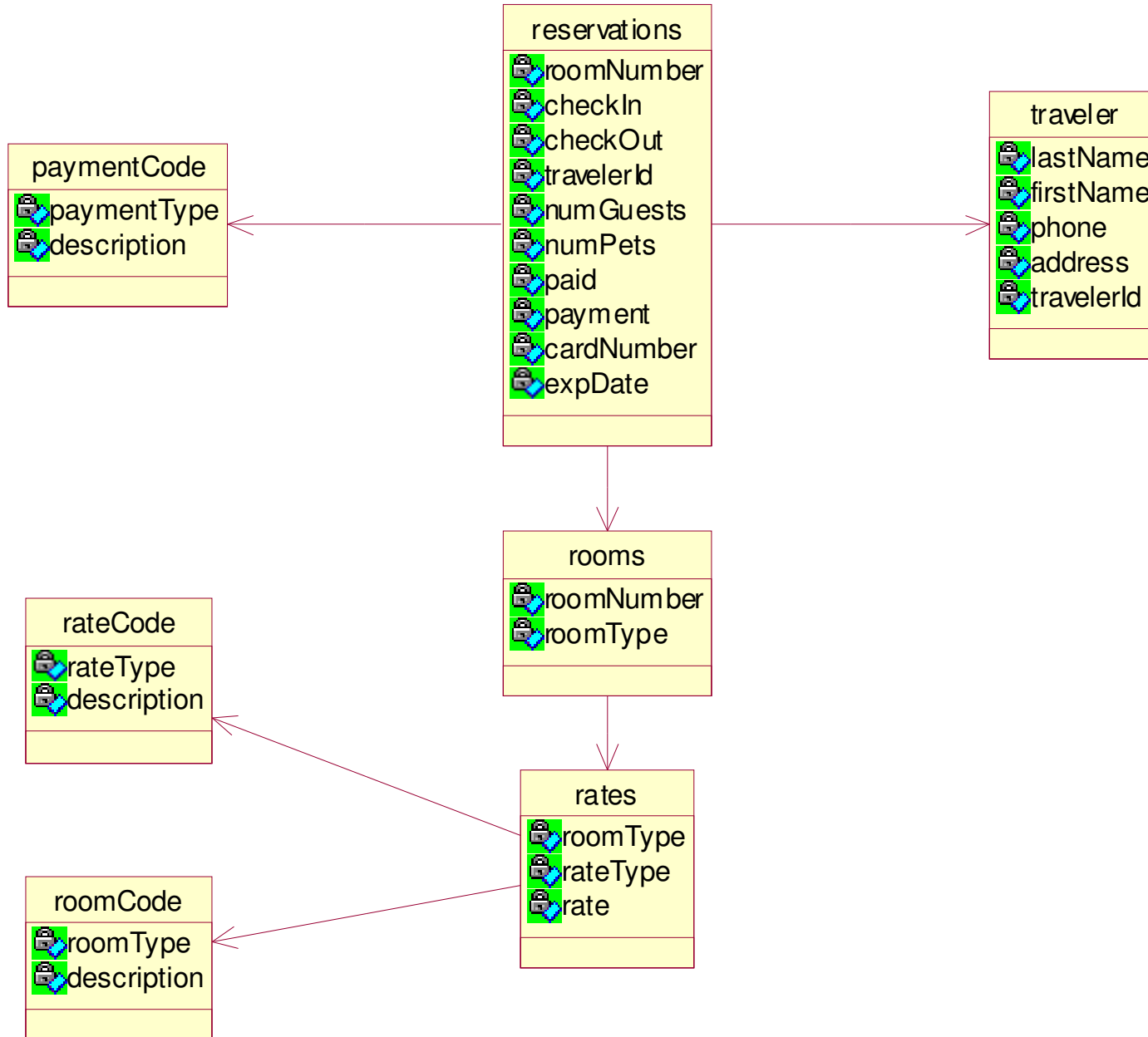
Credit Card Expiration Date (yyyy-mm) :

Done Internet

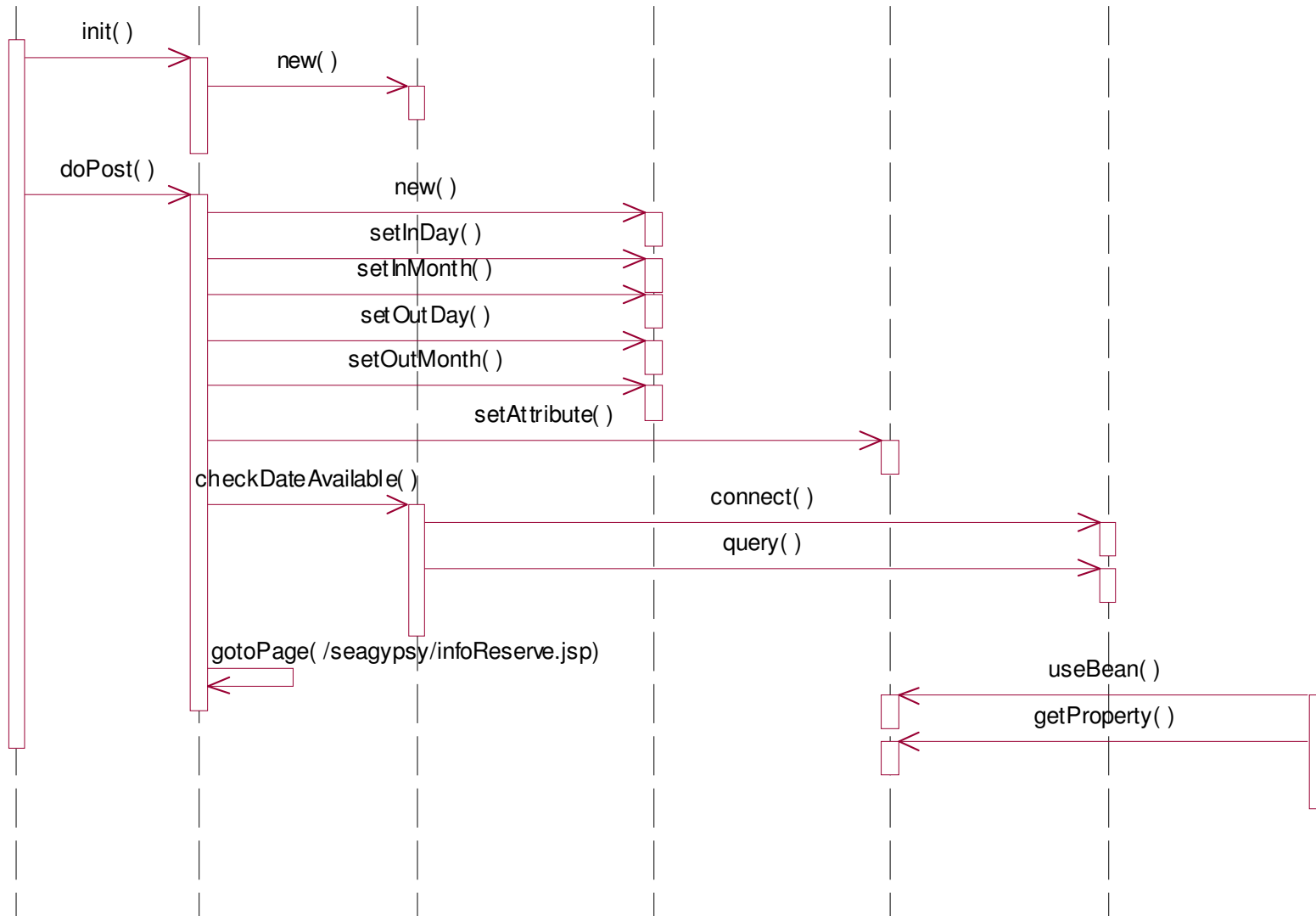
UML Class Diagram



UML Database Schema



UML Sequence Diagram



Design Issues

- Good software design separates the presentation from the business logic from the data model.
- Model/View/Controller, Mediator, and Façade design patterns can be used to improve maintainability and distribution of tasks

Design Issues

- Web applications are often developed in a group whose members have a combination of skills
 - System Administrator (Unix, Apache, Tomcat)
 - Web Developer (HTML, Graphics Design)
 - Software Developer (Java)

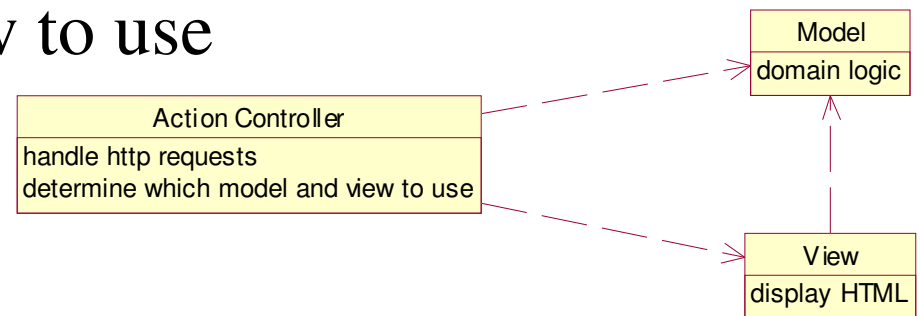
Design Issues

- Object-Oriented approach facilitates reuse, encapsulation, distribution of responsibilities.
- Presentation layer (view) can primarily be restricted to JSP.
- Business logic (controller) can primarily be restricted to Java using Servlets.
- Data model (model) may be split between DB and Java.
 - Domain model objects don't care if they are being used in a web application or a stand-alone application.

Web Application Patterns

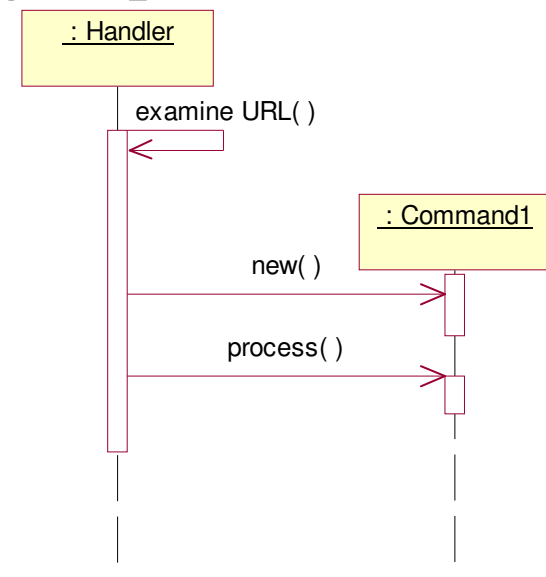
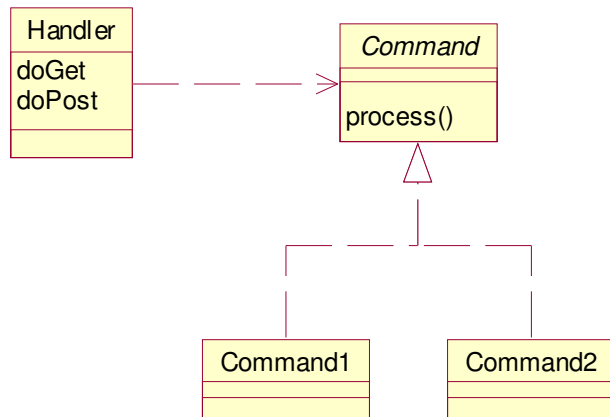
- Action Controller

- Each web page has a controller object that knows which model and view to use



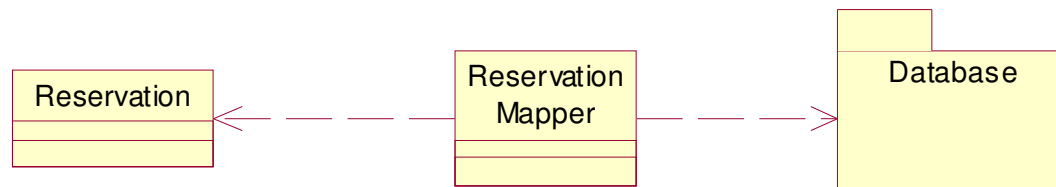
- Front Controller

- A single object handles incoming requests and passes them to other objects



Web Application Patterns

- Data Mapper
 - Each Domain Object has a Mapper object that knows how to store and retrieve it from the DB



- See more at:
 - <http://martinfowler.com/isa/>
- For a framework that contains many patterns:
 - See Espresso at <http://www.jcorporate.com/>

Use in curriculum

- Software Design and Development
- Networks
 - Web Application becomes client of a C++ server
- Senior Seminars
 - On-line newspaper
 - Course Evaluation system

Resources

- Core Servlets and JavaServer Pages
 - by Marty Hall
- Core Web Programming
 - by Marty Hall and Larry Brown
- <http://java.sun.com>
- <http://java.sun.com/webservices/docs/ea2/tutorial/index.html>
- <http://www.wantjava.com/resources.jsp>
- <http://www.servlets.com>
- <http://jakarta.apache.org>