Core Servlets and JavaServer Pages / 2e
Volume 1: Core Technologies
Marty Hall • Larry Brown

Handling the Client Request: Form Data

Agenda

- The role of form data
- Creating and submitting HTML forms
- Reading individual request parameters
- Reading the entire set of request parameters
- Handling missing and malformed data
- Dealing with incomplete form submissions
- Filtering special characters out of the request parameters

The Role of Form Data

- **Example URL at online travel agent**
  - http://host/path?user=Marty+Hall&origin=bwi&dest=lax
  - Names come from HTML author; values from end user
- **Parsing form (query) data in traditional CGI**
  - Read the data one way (QUERY_STRING) for GET requests, another way (standard input) for POST requests
  - Chop pairs at ampersands, then separate parameter names (left of the =) from parameter values (right of the =)
  - URL decode values (e.g., "%7E" becomes "~")
- **Greatly simplified in servlets**
  - Use request.getParameter in all cases.
  - Gives URL-decoded result

The Role of Form Data

- GET requests attach the data to the end of the URL after a question mark
- POST requests send the data on a separate line (better for security but can be “sniffed” since in plain text)

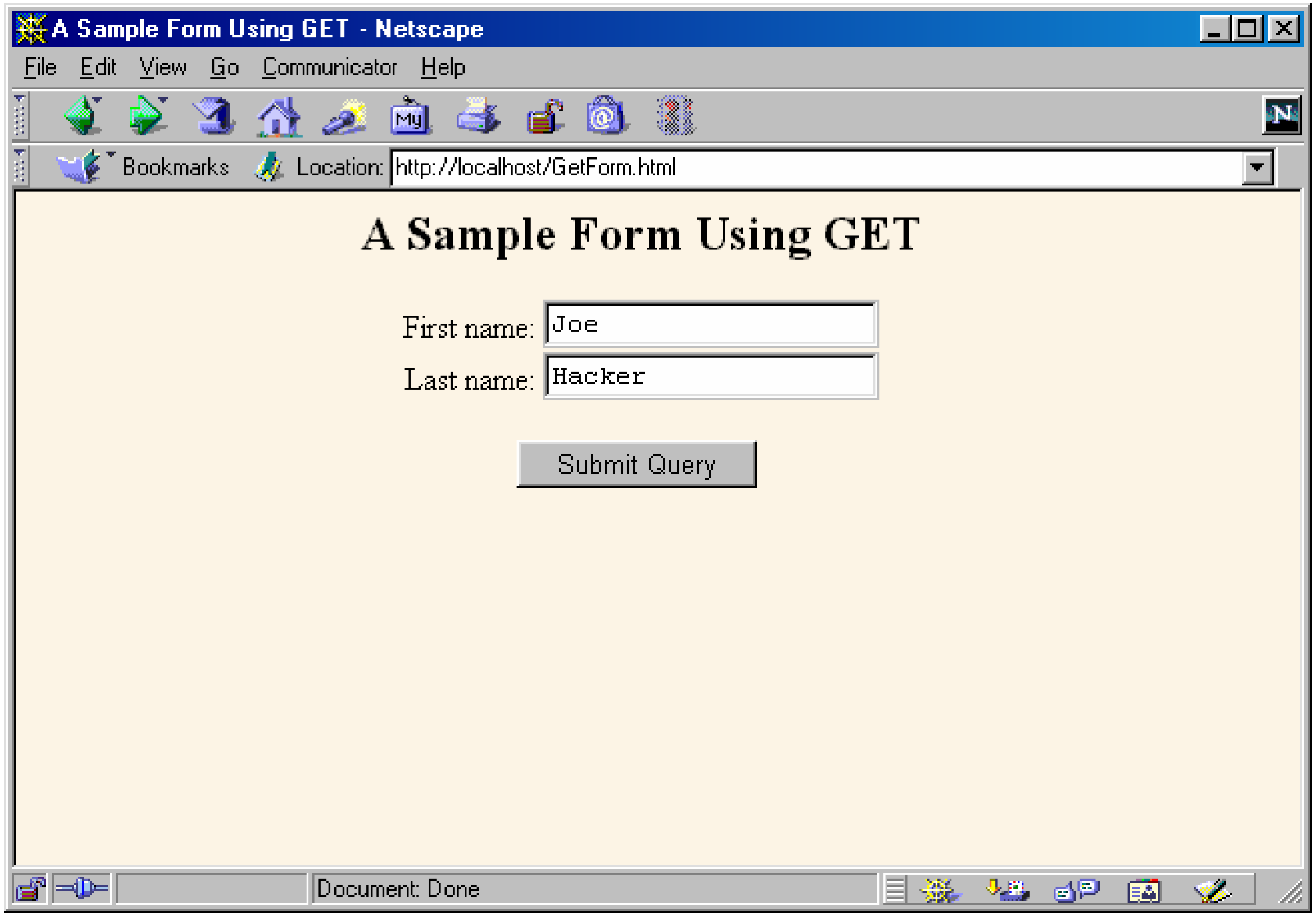
Creating Form Data: HTML Forms

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>A Sample Form Using GET</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">A Sample Form Using GET</H2>

<FORM ACTION="http://localhost:8088/SomeProgram">
  <CENTER>
  First name:
  <INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
  Last name:
  <INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
  <INPUT TYPE="SUBMIT"> <!-- Press this to submit form -->
  </CENTER>
</FORM>
</BODY></HTML>
```

Aside: Installing HTML Files

- **HTML files do not go in WEB-INF/classes**
  - They go in directory that *contains* WEB-INF (C:\jakarta-tomcat-5.5.9\webapps\ROOT)
- **Tomcat**
  - *install_dir*\webapps\ROOT\Form.html or
  - *install_dir*\webapps\ROOT\SomeDir\Form.html
- **URL**
  - http://localhost/Form.html or
  - http://localhost/*SomeDir*/Form.html
- **Custom Web applications**
  - Use a different dir with the same structure as the default Web app
  - Use directory name in URL (http://host/*dirName*/…)
  - For details, see Section 2.11 of *Core Servlets & JSP* (2nd Ed) and Chapter 4 of *More Servlets & JSP*

HTML Form: Initial Result


A Sample Form Using GET - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: http://localhost/GetForm.html

A Sample Form Using GET

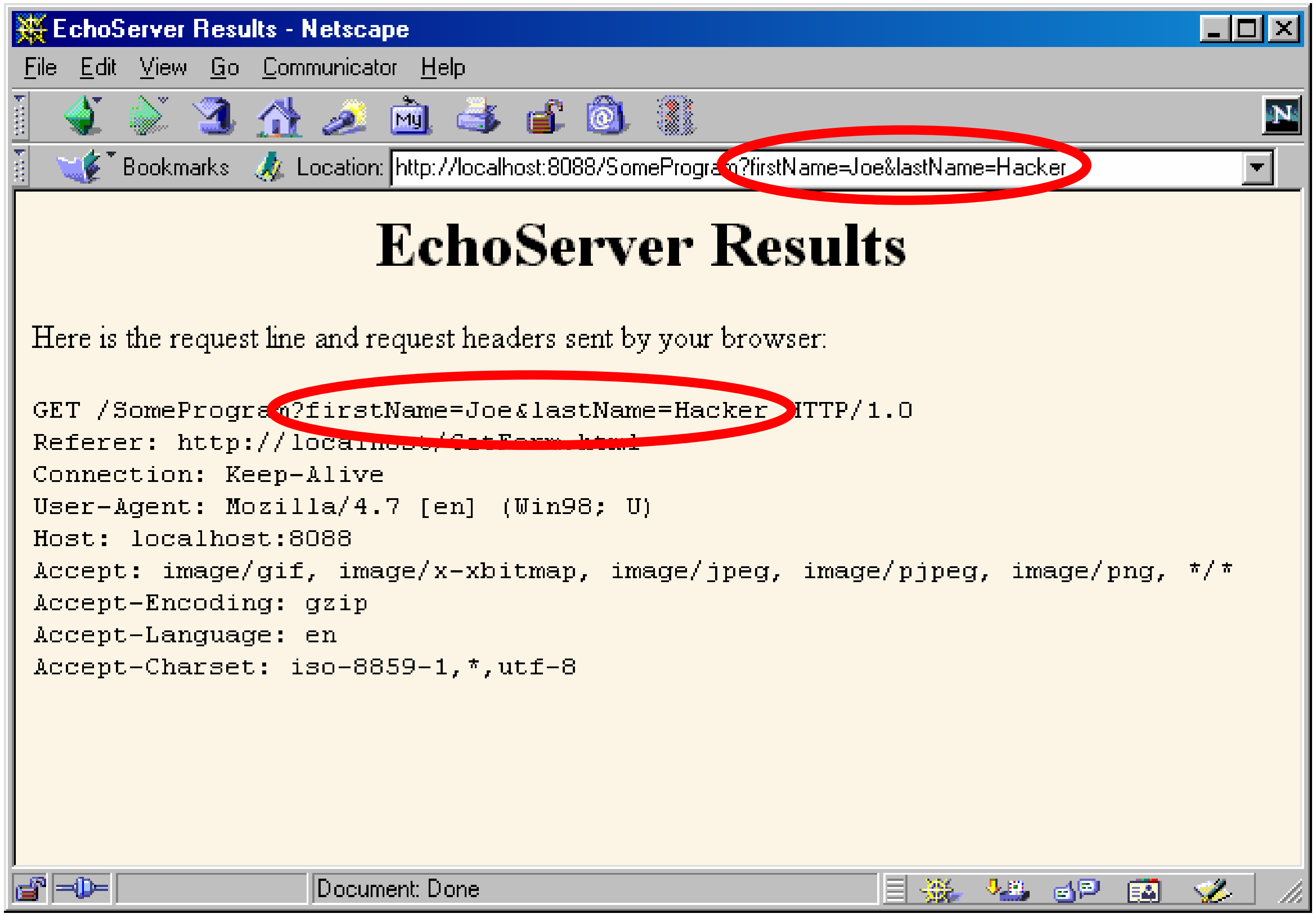
First name:	Joe
Last name:	Hacker

Submit Query
Document: Done

HTML Form: Submission Result (Data Sent to EchoServer)

- Echoserver is a Java application that acts as a mini webserver
- Download the jar file from http://volume1.coreservlets.com/archive/
- Compile and run it
- Open your browser and, when you enter the URL, make sure to include port 8088 with localhost - **`http://localhost:8088/servlet/coreservlets.SomeProgram?firstName=Joe&SecondName=Hacker`**

HTML Form: Submission Result (Data Sent to EchoServer)


EchoServer Results - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: http://localhost:8088/SomeProgram?firstName=Joe&lastName=Hacker
Document: Done

EchoServer Results

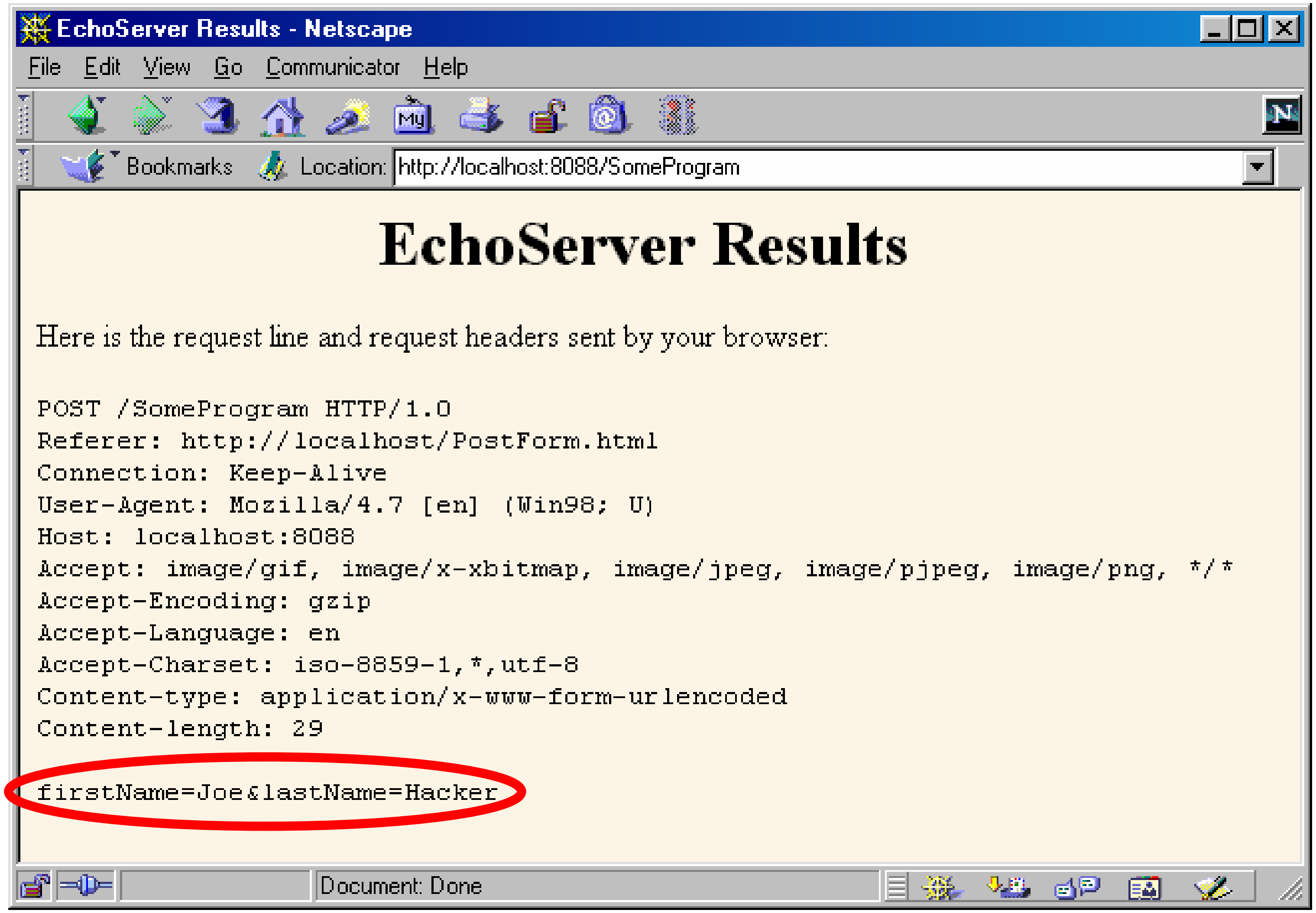
Here is the request line and request headers sent by your browser:

```
GET /SomeProgram?firstName=Joe&lastName=Hacker HTTP/1.0
Referer: http://localhost/GetForm.html
Connection: Keep-Alive
User-Agent: Mozilla/4.7 [en] (Win98; U)
Host: localhost:8088
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
```

Sending POST Data

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
   Transitional//EN">
<HTML>
<HEAD><TITLE>A Sample Form Using POST</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">A Sample Form Using POST</H2>

<FORM ACTION="http://localhost:8088/SomeProgram"
      METHOD="POST">
  <CENTER>
  First name:
  <INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
  Last name:
  <INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
  <INPUT TYPE="SUBMIT">
  </CENTER>
</FORM>

</BODY></HTML>
```

Sending POST Data


EchoServer Results - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: http://localhost:8088/SomeProgram
Document: Done

EchoServer Results

Here is the request line and request headers sent by your browser:

```
POST /SomeProgram HTTP/1.0
Referer: http://localhost/PostForm.html
Connection: Keep-Alive
User-Agent: Mozilla/4.7 [en] (Win98; U)
Host: localhost:8088
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
Content-type: application/x-www-form-urlencoded
Content-length: 29

firstName=Joe&lastName=Hacker
```

Reading Form Data In Servlets

- ***request.getParameter("name")***
  - Returns URL-decoded value of first occurrence of name in query string
  - Works identically for GET and POST requests
  - Returns null if no such parameter is in query data
- ***request.getParameterValues("name")***
  - Returns an array of the URL-decoded values of *all* occurrences of name in query string
  - Returns a one-element array if param not repeated
  - Returns null if no such parameter is in query
  - Needed for multi-selectable list boxes which repeat the parameter name for each selected element in the list
- **Values supplied to either method are case sensitive**

Reading Form Data In Servlets

- **request.getParameterNames()**
  - Returns Enumeration or Map of request parameters
  - Entry must be cast to a String and used in a getParameter or getParameterValues call
  - Usually reserved for debugging
  - Don’t assume parameters are passed in any particular order
  - Also useful when parameter names are dynamic and/or could have meaning to the program
    e.g., row-1-col-3-value

Handling Input in Multiple Languages

- Use server's default character set

```
String firstName =
  request.getParameter("firstName");
```

- Convert from English (Latin-1) to Japanese

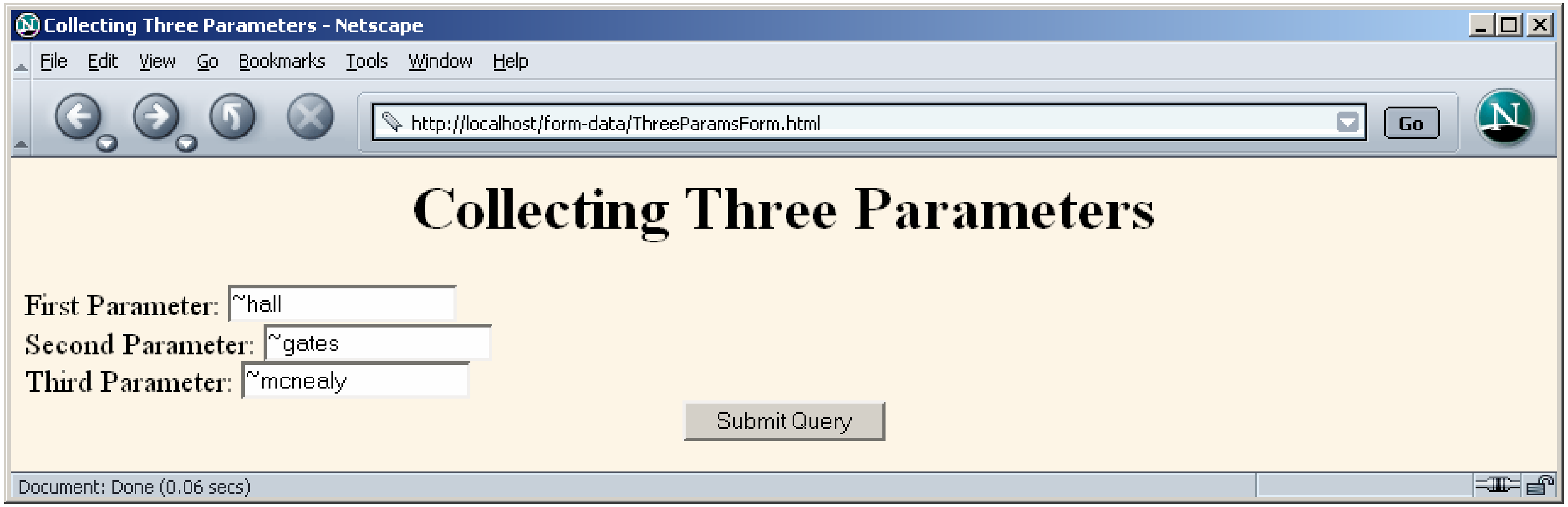
```
String firstNameWrongEncoding =
  request.getParameter("firstName");
String firstName =
  new String(firstNameWrongEncoding.getBytes(),
                        "Shift_JIS");
```

- Accept either English or Japanese

```
request.setCharacterEncoding("JISAutoDetect");
String firstName =
  request.getParameter("firstName");
```

An HTML Form With Three Parameters

```
<FORM ACTION="/servlet/coreservlets.ThreeParams">
  First Parameter:  <INPUT TYPE="TEXT" NAME="param1"><BR>
  Second Parameter: <INPUT TYPE="TEXT" NAME="param2"><BR>
  Third Parameter:  <INPUT TYPE="TEXT" NAME="param3"><BR>
  <CENTER><INPUT TYPE="SUBMIT"></CENTER>
</FORM>
```


Collecting Three Parameters - Netscape
File Edit View Go Bookmarks Tools Window Help
http://localhost/form-data/ThreeParamsForm.html
Go

Collecting Three Parameters

First Parameter: ~hall
Second Parameter: ~gates
Third Parameter: ~mcnealy

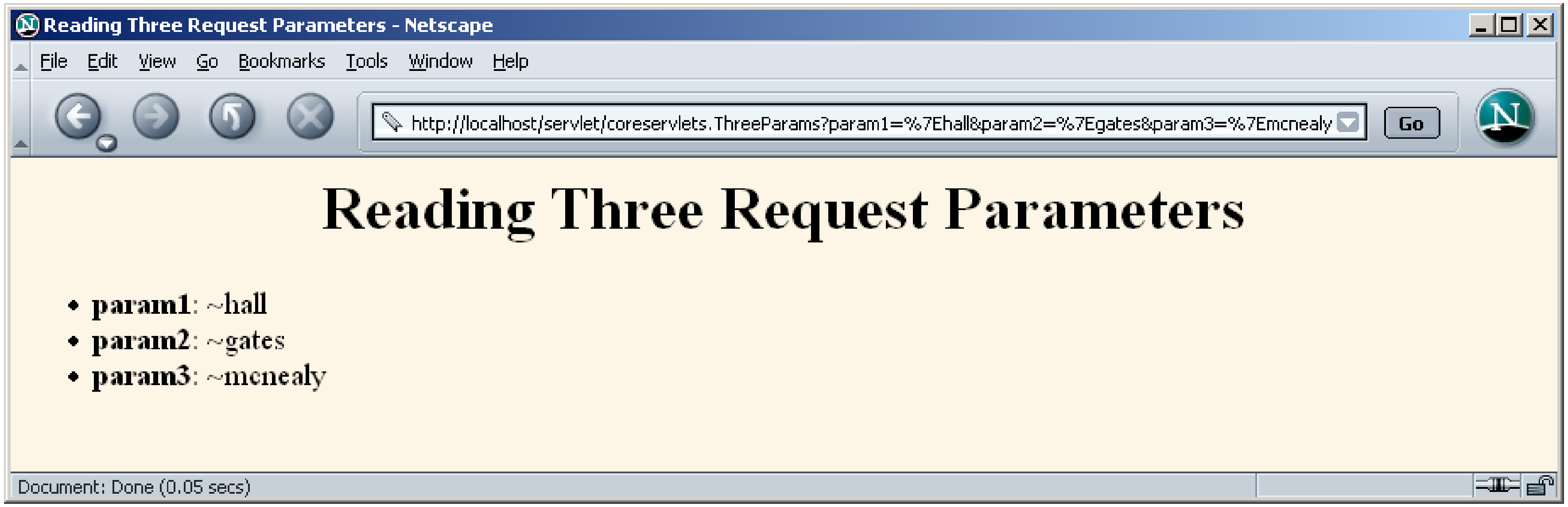
Submit Query

Document: Done (0.06 secs)

Reading the Three Parameters

```
public class ThreeParams extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String title = "Reading Three Request Parameters";
    String docType =
      "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 " +
      "Transitional//EN\">\n";
    out.println(docType +
                "<HTML>\n" +
                "<HEAD><TITLE>"+title + "</TITLE></HEAD>\n" +
                "<BODY BGCOLOR=\"#FDF5E6\">\n" +
                "<H1 ALIGN=\"CENTER\">" + title + "</H1>\n" +
                "<UL>\n" +
                "  <LI><B>param1</B>: "
                + request.getParameter("param1") + "\n" +
                "  <LI><B>param2</B>: "
                + request.getParameter("param2") + "\n" +
                "  <LI><B>param3</B>: "
                + request.getParameter("param3") + "\n" +
                "</UL>\n" +
                "</BODY></HTML>");
  }
}
```

Reading Three Parameters: Result


Reading Three Request Parameters - Netscape
File Edit View Go Bookmarks Tools Window Help
Go

http://localhost/servlet/coreservlets.ThreeParams?param1=%7Ehall&param2=%7Egates&param3=%7Emcnealy

Reading Three Request Parameters

- **param1**: ~hall
- **param2**: ~gates
- **param3**: ~mcnealy

Document: Done (0.05 secs)

Reading All Parameters

- **Java provides the ability to look up all the parameter names that are sent and puts their values in a table**
- **Shows which have missing values and those having multiples values**
- **Uses getParamaterNames which returns and Enumeration**
  - Uses hasMoreElements and nextElement to iterate through the list
  - Cast returned value to a String from Object before using it in getParamaterValues
    - Returned array of String can be empty, contain a single value or multiple values

Reading All Parameters

```
public class ShowParameters extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String docType =
    "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 " +
    "Transitional//EN\">\n";
    String title = "Reading All Request Parameters";
    out.println(docType +
                "<HTML>\n" +
                "<HEAD><TITLE>"+title + "</TITLE></HEAD>\n"+
                "<BODY BGCOLOR=\"#FDF5E6\">\n" +
                "<H1 ALIGN=CENTER>" + title + "</H1>\n" +
                "<TABLE BORDER=1 ALIGN=CENTER>\n" +
                "<TR BGCOLOR=\"#FFAD00\">\n" +
                "<TH>Parameter Name<TH>Parameter Value(s)");
```

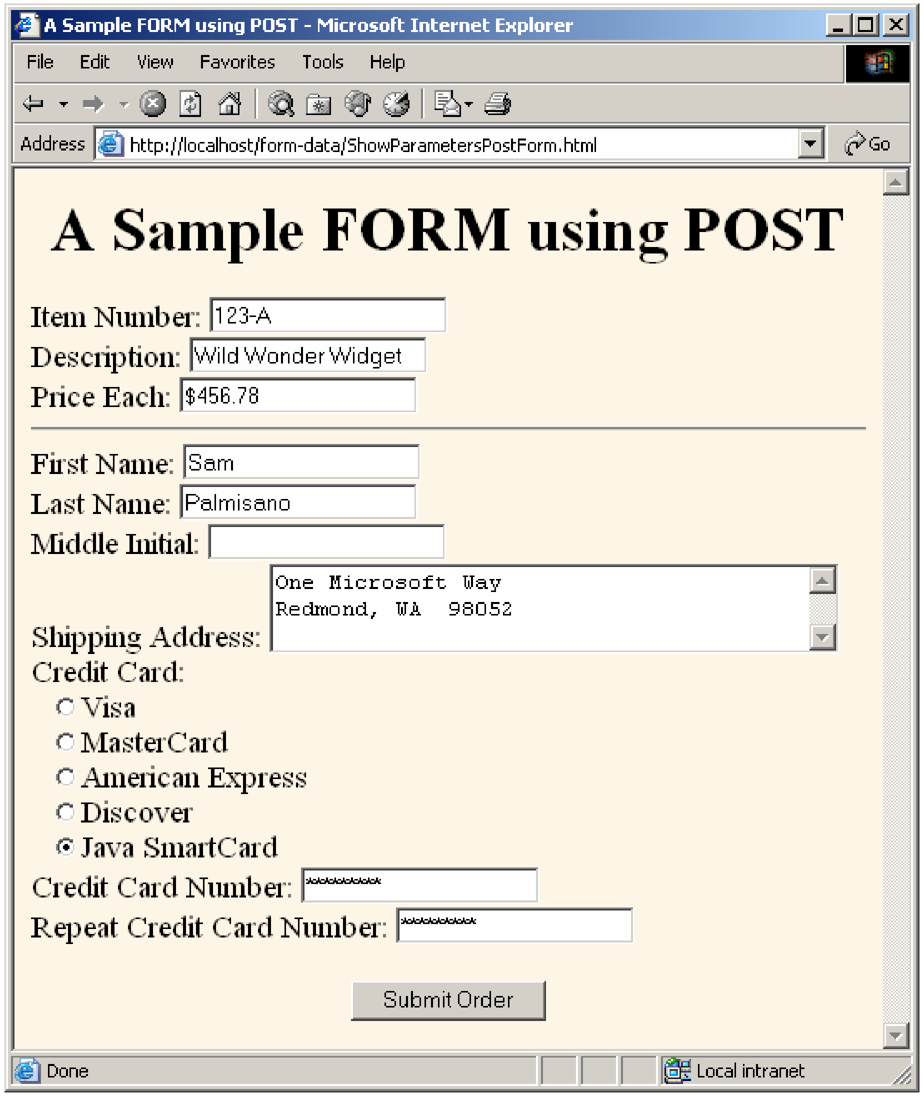
Reading All Parameters (Continued)

```
    Enumeration paramNames = request.getParameterNames();
    while(paramNames.hasMoreElements()) {
      String paramName = (String)paramNames.nextElement();
      out.print("<TR><TD>" + paramName + "\n<TD>");
      String[] paramValues =
        request.getParameterValues(paramName);
      if (paramValues.length == 1) {
        String paramValue = paramValues[0];
        if (paramValue.length() == 0)
          out.println("<I>No Value</I>");
        else
          out.println(paramValue);
      } else {
        out.println("<UL>");
        for(int i=0; i<paramValues.length; i++) {
          out.println("<LI>" + paramValues[i]);
        }
        out.println("</UL>");
      }
    }
    out.println("</TABLE>\n</BODY></HTML>");
  }
```

Reading All Parameters (Continued)

```
  public void doPost(HttpServletRequest request,
                     HttpServletResponse response)
      throws ServletException, IOException {
    doGet(request, response);
  }
}
```

Reading All Parameters (Sample Form)


A Sample FORM using POST - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Go
Done
Local intranet

http://localhost/form-data/ShowParametersPostForm.html

A Sample FORM using POST

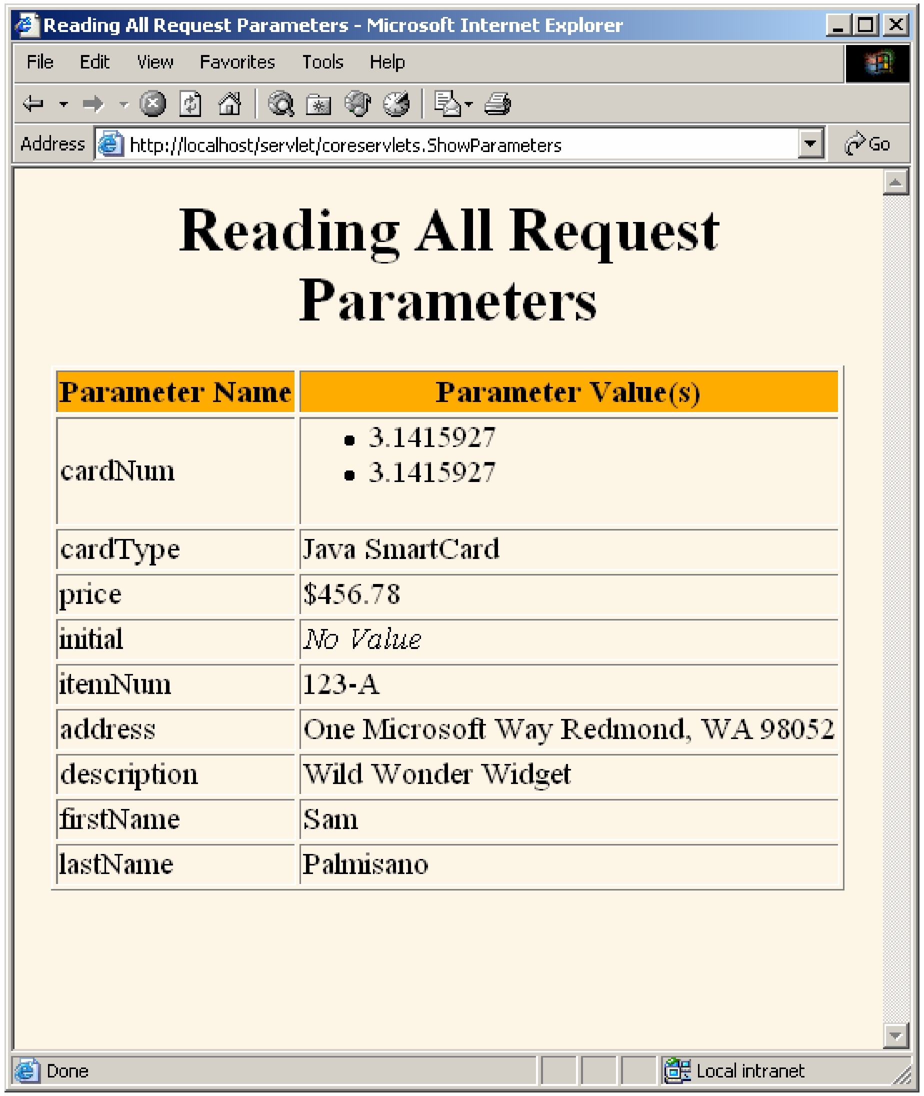
Item Number: 123-A
Description: Wild Wonder Widget
Price Each: $456.78

First Name: Sam
Last Name: Palmisano
Middle Initial:
Shipping Address:
```
One Microsoft Way
Redmond, WA  98052
```
Credit Card:
- ○ Visa
- ○ MasterCard
- ○ American Express
- ○ Discover
- ● Java SmartCard

Credit Card Number: **********
Repeat Credit Card Number: **********

Submit Order

Reading All Parameters (Result)


Reading All Request Parameters - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/servlet/coreservlets.ShowParameters
Go
Done
Local intranet

Reading All Request Parameters

Parameter Name	Parameter Value(s)
cardNum	• 3.1415927 • 3.1415927
cardType	Java SmartCard
price	$456.78
initial	*No Value*
itemNum	123-A
address	One Microsoft Way Redmond, WA 98052
description	Wild Wonder Widget
firstName	Sam
lastName	Palmisano

Checking for Missing and Malformed Data

- **Missing**
  - Field missing in form
    - getParameter returns null
  - Field blank when form submitted
    - getParameter returns an empty string (or possibly a string with whitespace in it)
  - Must check for null before checking for empty string

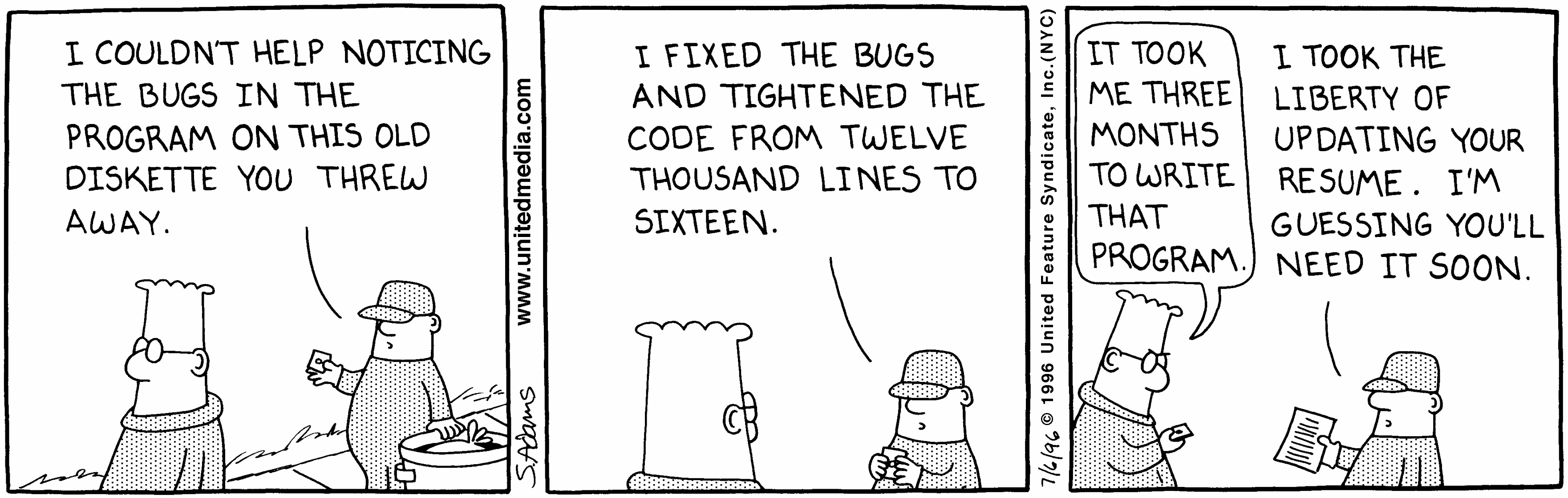
```
String param = request.getParameter("someName");
if ((param == null) || (param.trim().equals(""))) {
  doSomethingForMissingValues(...);
} else {
  doSomethingWithParameter(param);
}
```

- **Malformed**
  - Value is a nonempty string in the wrong format

Handling Missing and Malformed Data

- **Use default values**
  - Replace missing values with application-specific standard values
- **Redisplay the form**
  - Show the form again, with missing values flagged
  - Previously-entered values should be preserved
  - Four options to implement this
    - Have the same servlet present the form, process the data, and present the results.
    - Have one servlet present the form; have a second servlet process the data and present the results.
    - Have a JSP page “manually” present the form; have a servlet or JSP page process the data and present the results.
    - Have a JSP page present the form, automatically filling in the fields with values obtained from a data object. Have a servlet or JSP page process the data and present the results

Example of Using Default Values: A Résumé-Posting Site


I COULDN'T HELP NOTICING THE BUGS IN THE PROGRAM ON THIS OLD DISKETTE YOU THREW AWAY.
www.unitedmedia.com
S.Adams
I FIXED THE BUGS AND TIGHTENED THE CODE FROM TWELVE THOUSAND LINES TO SIXTEEN.
7/6/96 © 1996 United Feature Syndicate, Inc.(NYC)
IT TOOK ME THREE MONTHS TO WRITE THAT PROGRAM.
I TOOK THE LIBERTY OF UPDATING YOUR RESUME. I'M GUESSING YOU'LL NEED IT SOON.

Reading All Parameters (Sample Form)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML><HEAD><TITLE>Free Resume Posting</TITLE>
<LINK REL=STYLESHEET HREF="jobs-site-styles.css“
TYPE="text/css">
</HEAD>
<BODY>
<H1>hot-computer-jobs.com</H1>
<P CLASS="LARGER">
To use our <I>free</I> resume-posting service, simply fill
out the brief summary of your skills below. Use "Preview"
to check the results, then press "Submit" once it is
ready. Your mini-resume will appear online within 24
hours.</P>
<HR>

<FORM ACTION="/servlet/coreservlets.SubmitResume"
      METHOD="POST">
```

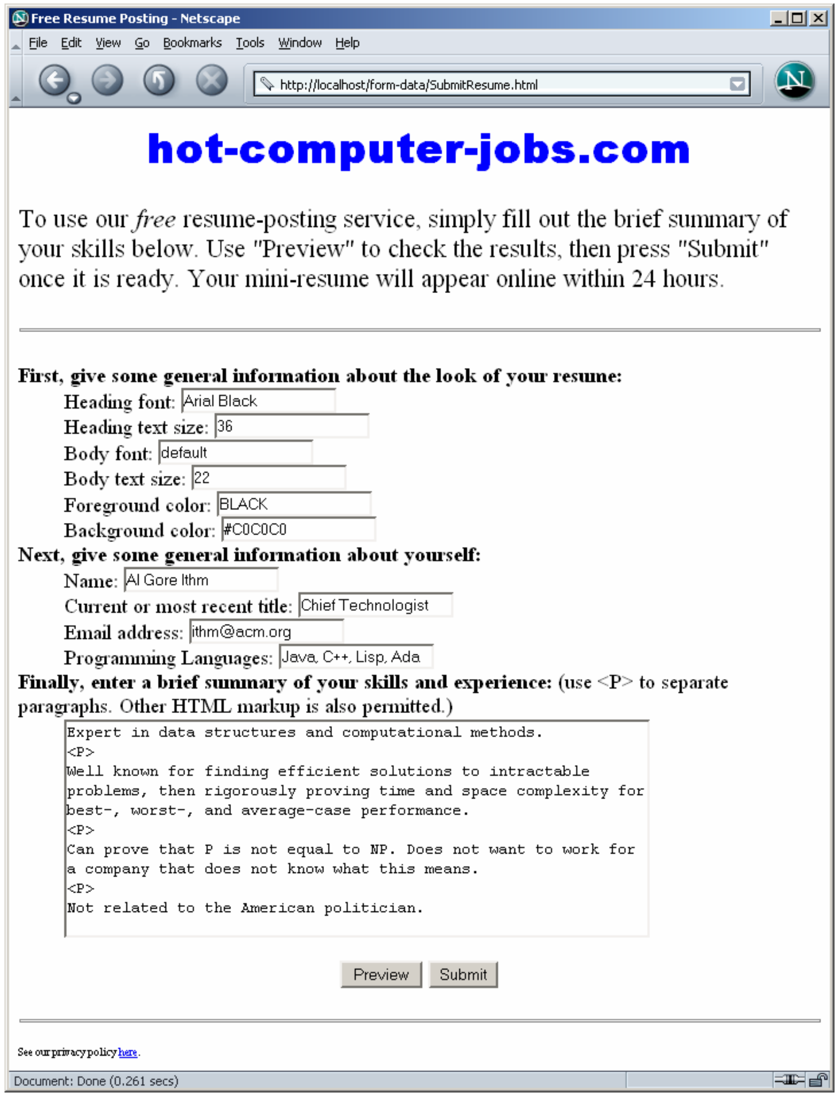
Reading All Parameters (Sample Form)

```
<DL>
<DT><B>First, give some general information about the
look of your resume:</B>
<DD>Heading font:
    <INPUT TYPE="TEXT" NAME="headingFont"VALUE="default">
<DD>Heading text size:
    <INPUT TYPE="TEXT" NAME="headingSize" VALUE=32>
<DD>Body font:
    <INPUT TYPE="TEXT" NAME="bodyFont" VALUE="default">
<DD>Body text size:
    <INPUT TYPE="TEXT" NAME="bodySize" VALUE=18>
<DD>Foreground color:
    <INPUT TYPE="TEXT" NAME="fgColor" VALUE="BLACK">
<DD>Background color:
    <INPUT TYPE="TEXT" NAME="bgColor" VALUE="WHITE">
```

Reading All Parameters (Sample Form)

```
<DT><B>Next, give some general information about yourself:</B>
<DD>Name: <INPUT TYPE="TEXT" NAME="name">
<DD>Current or most recent title:
    <INPUT TYPE="TEXT" NAME="title">
<DD>Email address: <INPUT TYPE="TEXT" NAME="email">
<DD>Programming Languages: <INPUT TYPE="TEXT" NAME="languages">

<DT><B>Finally, enter a brief summary of your skills and
    experience:</B> (use &lt;P&gt; to separate paragraphs.
    Other HTML markup is also permitted.)

<DD><TEXTAREA NAME="skills" ROWS=10 COLS=60 WRAP="SOFT"></TEXTAREA>
</DL>
  <CENTER>
    <INPUT TYPE="SUBMIT" NAME="previewButton" Value="Preview">
    <INPUT TYPE="SUBMIT" NAME="submitButton" Value="Submit">
  </CENTER>
</FORM>
<HR>
<P CLASS="TINY">See our privacy policy
<A HREF="we-will-spam-you.html">here</A>.</P>
</BODY></HTML>
```

Résumé-Posting Site: Input Form


Free Resume Posting - Netscape
File Edit View Go Bookmarks Tools Window Help
http://localhost/form-data/SubmitResume.html

hot-computer-jobs.com

To use our *free* resume-posting service, simply fill out the brief summary of your skills below. Use "Preview" to check the results, then press "Submit" once it is ready. Your mini-resume will appear online within 24 hours.

First, give some general information about the look of your resume:

Heading font: Arial Black
Heading text size: 36
Body font: default
Body text size: 22
Foreground color: BLACK
Background color: #C0C0C0

Next, give some general information about yourself:

Name: Al Gore Ithm
Current or most recent title: Chief Technologist
Email address: ithm@acm.org
Programming Languages: Java, C++, Lisp, Ada

Finally, enter a brief summary of your skills and experience: (use <P> to separate paragraphs. Other HTML markup is also permitted.)

```
Expert in data structures and computational methods.
<P>
Well known for finding efficient solutions to intractable
problems, then rigorously proving time and space complexity for
best-, worst-, and average-case performance.
<P>
Can prove that P is not equal to NP. Does not want to work for
a company that does not know what this means.
<P>
Not related to the American politician.
```

See our privacy policy here.

Résumé-Posting Site: Servlet Code

- **Servlet variables**
  - **`name, title, email, language, skills`**
    - Missing values are replaced by default values
  - **`fgColor, bgColor`**
    - Colors of the background and foreground
      - Default is white and black
  - **`headingFont, bodyFont`**
    - font to be used for the heading and main body text
  - **`headingSize, bodySize`**
    - Point size for the main heading and body
    - Missing or nonnumeric values default to 32 and 18 respectively
    - Uses Integer.parseInt and try/catch for NumberFormatException to handle bad values

Résumé-Posting Site: Servlet Code

- **Servlet processing**
  - Once the servlet has good values for the variables, it builds a CSS out of them and embeds it directly in the page via the STYLE element
  - Name, job title and email address are centered under each other at the top of the page and the headingFont is used
  - Email address is placed inside a mailto: hyperlink so the employer can click on it to contact the applicant
  - The programming languages are determined by parsing the parameter **`languages`** via StringTokenizer and placed in a bullet list
  - The text from the **`skills`** parameter is placed at the bottom of the page

Résumé-Posting Site: Servlet Code

```
public void doPost(HttpServletRequest request,
  HttpServletResponse response)
  throws ServletException, IOException {

   response.setContentType("text/html");
   PrintWriter out = response.getWriter();

   if (request.getParameter("previewButton")
         != null)
      { showPreview(request, out);}
   else
    { storeResume(request);
      showConfirmation(request, out);
    }
  }
```

Résumé-Posting Site: Servlet Code

```
private void showPreview(HttpServletRequest request,PrintWriter out) {
   String headingFont = request.getParameter("headingFont");
   headingFont = replaceIfMissingOrDefault(headingFont, "");

   int headingSize = getSize(request.getParameter("headingSize"), 32);

   String bodyFont = request.getParameter("bodyFont");
   bodyFont = replaceIfMissingOrDefault(bodyFont, "");

   int bodySize = getSize(request.getParameter("bodySize"), 18);

   String fgColor = request.getParameter("fgColor");
   fgColor = replaceIfMissing(fgColor, "BLACK");
   String bgColor = request.getParameter("bgColor");
   bgColor = replaceIfMissing(bgColor, "WHITE");

   String name = request.getParameter("name");
   name = replaceIfMissing(name, "Lou Zer");
   String title = request.getParameter("title");
   title = replaceIfMissing(title, "Loser");
   String email = request.getParameter("email");
   email = replaceIfMissing(email, "contact@hot-computer-jobs.com");
   String languages = request.getParameter("languages");
   languages = replaceIfMissing(languages, "<I>None</I>");
   String languageList = makeList(languages);
   String skills = request.getParameter("skills");
   skills = replaceIfMissing(skills, "Not many, obviously.");
```

Résumé-Posting Site: Servlet Code

```
  out.println
    (ServletUtilities.DOCTYPE + "\n" +
     "<HTML><HEAD><TITLE>Resume for " + name + "</TITLE>\n"
 +
     makeStyleSheet(headingFont, headingSize,
                    bodyFont, bodySize,
                    fgColor, bgColor) + "\n" +
     "</HEAD>\n" +
     "<BODY>\n" +
     "<CENTER>\n"+
     "<SPAN CLASS=\"HEADING1\">" + name + "</SPAN><BR>\n" +
     "<SPAN CLASS=\"HEADING2\">" + title + "<BR>\n" +
     "<A HREF=\"mailto:" + email + "\">" + email +
         "</A></SPAN>\n" +
     "</CENTER><BR><BR>\n" +
     "<SPAN CLASS=\"HEADING3\">Programming Languages" +
     "</SPAN>\n" +
     makeList(languages) + "<BR><BR>\n" +
     "<SPAN CLASS=\"HEADING3\">Skills and Experience" +
     "</SPAN><BR><BR>\n" +
     skills + "\n" +
     "</BODY></HTML>");
} //end of ShowPreview
```

Résumé-Posting Site: Servlet Code (Continued)

```
private String makeStyleSheet(String headingFont,int heading1Size, String
   bodyFont, int bodySize, String fgColor, String bgColor) {
    int heading2Size = heading1Size*7/10;
    int heading3Size = heading1Size*6/10;
    String styleSheet =
    "<STYLE TYPE=\"text/css\">\n" + "<!--\n" +
    ".HEADING1 { font-size: " + heading1Size + "px;\n" + font-weight:
   bold;\n"    + " font-family: " + headingFont +"Arial, Helvetica, sans-
   serif;\n" + "}\n" +
   ".HEADING2 { font-size: " + heading2Size + "px;\n" + font-weight:
   bold;\n" + " font-family: " + headingFont + "Arial, Helvetica, sans-
   serif;\n" + "}\n" +
   ".HEADING3 { font-size: " + heading3Size + "px;\n" + font-weight:
   bold;\n" + " font-family: " + headingFont + "Arial, Helvetica, sans-
   serif;\n" + "}\n" +
   "BODY { color: " + fgColor + ";\n" + background-color: " + bgColor +
   ";\n" + "  font-size: " + bodySize + "px;\n" + font-family: " +
   bodyFont + "Times New Roman, Times, serif;\n" +
   "}\n" +
   "A:hover { color: red; }\n" + -->\n" +
   "</STYLE>";
   return(styleSheet);
  }
```

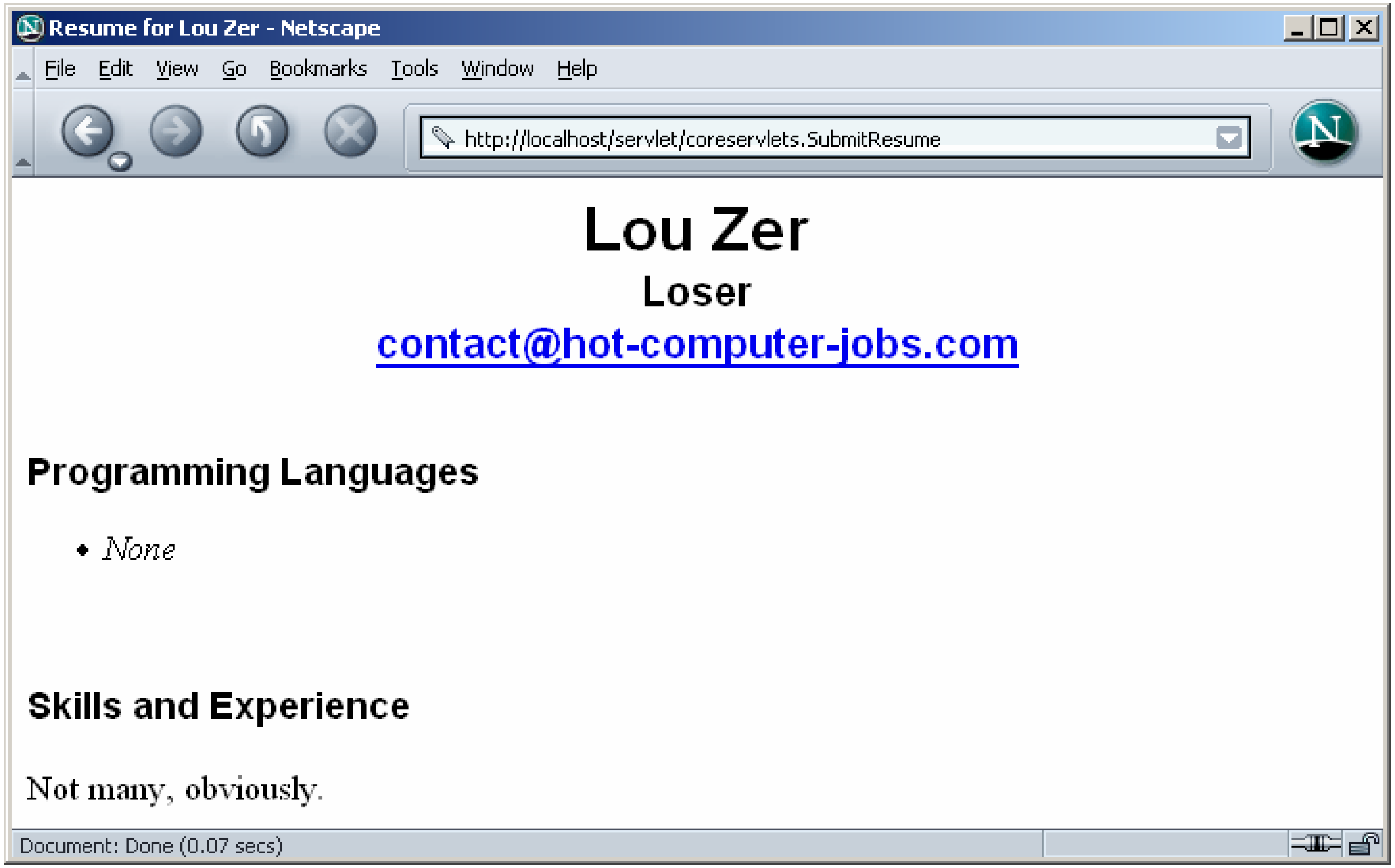
Résumé-Posting Site: Servlet Code (Continued)

```
private String replaceIfMissingOrDefault(String orig,
                                          String replacement) {
   if ((orig == null) ||
       (orig.trim().equals("")) ||
       (orig.equals("default"))) {
     return(replacement);
   } else {
     return(orig + ", ");
   }
 }

 // Takes a string representing an integer and returns it
 // as an int. Returns a default if the string is null
 // or in an illegal format.

 private int getSize(String sizeString, int defaultSize) {
   try {
     return(Integer.parseInt(sizeString));
   } catch(NumberFormatException nfe) {
     return(defaultSize);
   }
 }
```

Résumé-Posting Site: Servlet Code (Continued)

```
private String makeList(String listItems) {
   StringTokenizer tokenizer = new StringTokenizer(listItems, ", ");
   String list = "<UL>\n";
   while(tokenizer.hasMoreTokens()) {
     list = list + "  <LI>" + tokenizer.nextToken() + "\n";
   }
   list = list + "</UL>";
   return(list);
 }


 private void showConfirmation(HttpServletRequest request, PrintWriter out) {
   String title = "Submission Confirmed.";
   out.println(ServletUtilities.headWithTitle(title) +
               "<BODY>\n" +
               "<H1>" + title + "</H1>\n" +
               "Your resume should appear online within\n" +
               "24 hours. If it doesn't, try submitting\n" +
               "again with a different email address.\n" +
               "</BODY></HTML>");
 }
```

Résumé-Posting Site: Result for Incomplete Data


Resume for Lou Zer - Netscape
File Edit View Go Bookmarks Tools Window Help
http://localhost/servlet/coreservlets.SubmitResume
Document: Done (0.07 secs)

Lou Zer

Loser

contact@hot-computer-jobs.com

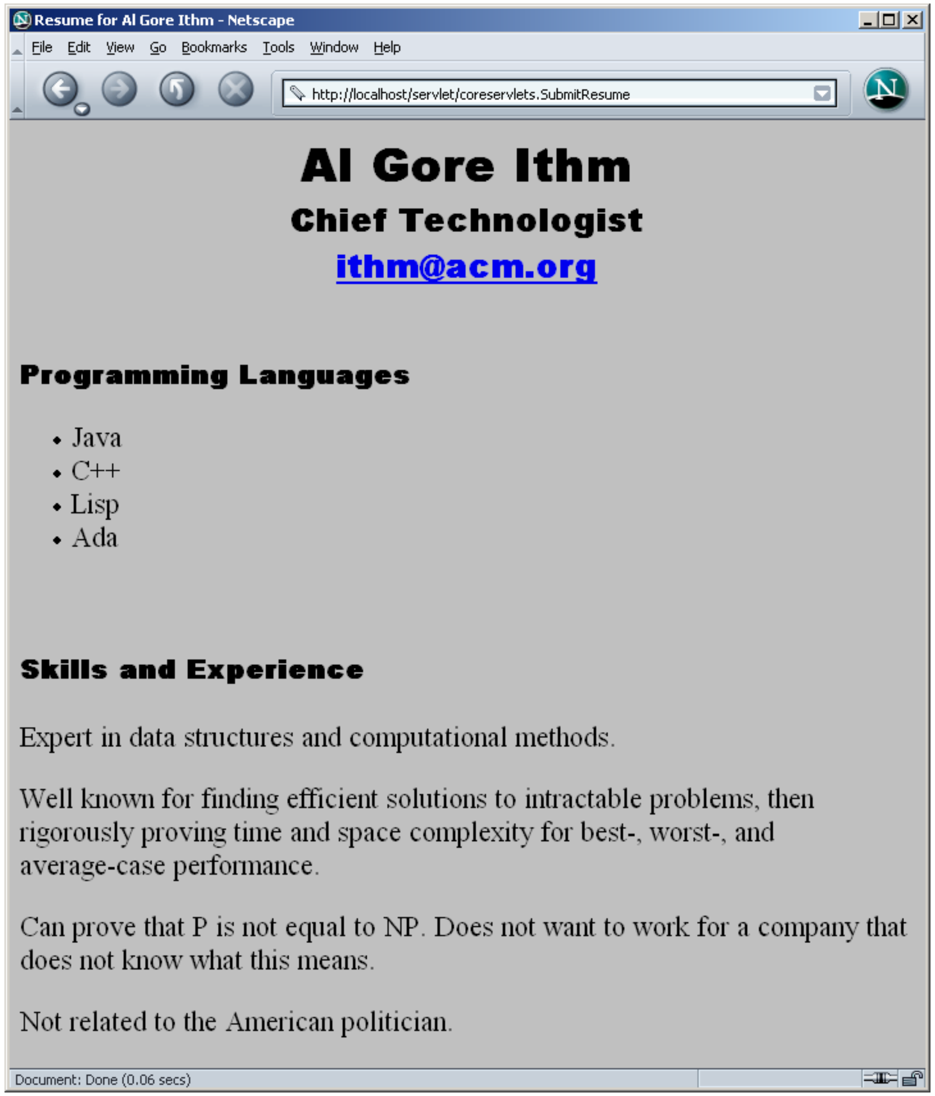
Programming Languages

- *None*

Skills and Experience

Not many, obviously.

Résumé-Posting Site: Result for Complete Data


Resume for Al Gore Ithm - Netscape
File Edit View Go Bookmarks Tools Window Help
http://localhost/servlet/coreservlets.SubmitResume
Document: Done (0.06 secs)

Al Gore Ithm

Chief Technologist

ithm@acm.org

Programming Languages

- Java
- C++
- Lisp
- Ada

Skills and Experience

Expert in data structures and computational methods.

Well known for finding efficient solutions to intractable problems, then rigorously proving time and space complexity for best-, worst-, and average-case performance.

Can prove that P is not equal to NP. Does not want to work for a company that does not know what this means.

Not related to the American politician.

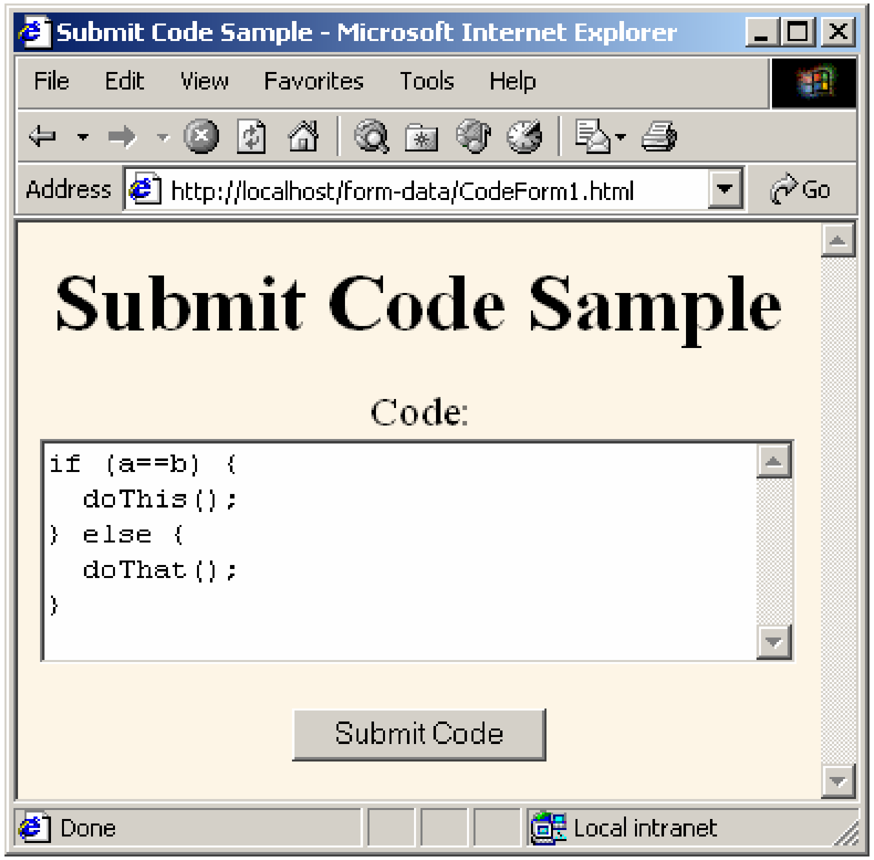
Filtering Strings for HTML-Specific Characters

- **You cannot safely insert arbitrary strings into servlet output**
  - < and > can cause problems anywhere
  - & and " can cause problems inside of HTML attributes
- **You sometimes cannot manually translate**
  - The string is derived from a program excerpt or another source where it is already in some standard format
  - The string is derived from HTML form data
- **Failing to filter special characters from form data makes you vulnerable to *cross-site scripting attack***
  - http://www.cert.org/advisories/CA-2000-02.html
  - http://www.microsoft.com/technet/security/topics/ExSumCS.asp

A Servlet that Displays Code Samples: No Filtering

```
public class BadCodeServlet extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    …
    out.println(docType +
                "<HTML>\n" +
                "<HEAD><TITLE>"+title+"</TITLE></HEAD>\n" +
                "<BODY BGCOLOR=\"#FDF5E6\">\n" +
                "<H1 ALIGN=\"CENTER\">" + title + "</H1>\n"+
                "<PRE>\n" +
                getCode(request) +
                "</PRE>\n" +
                "Now, wasn't that an interesting sample\n" +
                "of code?\n" +
                "</BODY></HTML>");
  }

  protected String getCode(HttpServletRequest request) {
    return(request.getParameter("code"));
  }
}
```

A Servlet that Displays Code Samples: No Special Chars

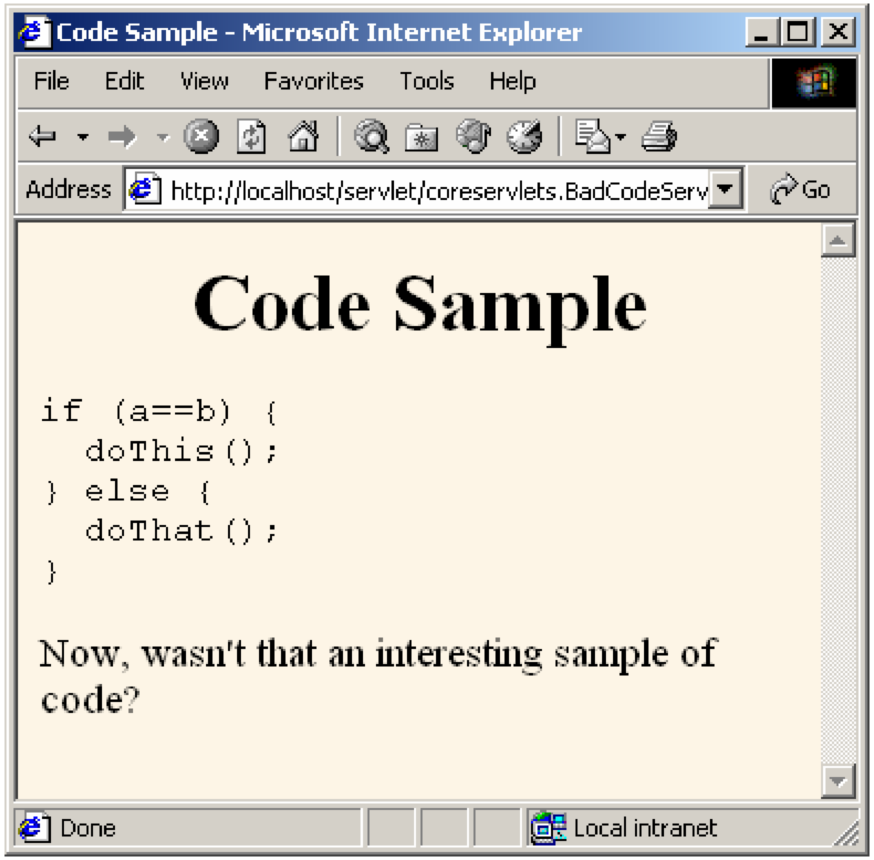

Submit Code Sample - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/form-data/CodeForm1.html
Go

Submit Code Sample

Code:

```
if (a==b) {
  doThis();
} else {
  doThat();
}
```

Submit Code
Done
Local intranet

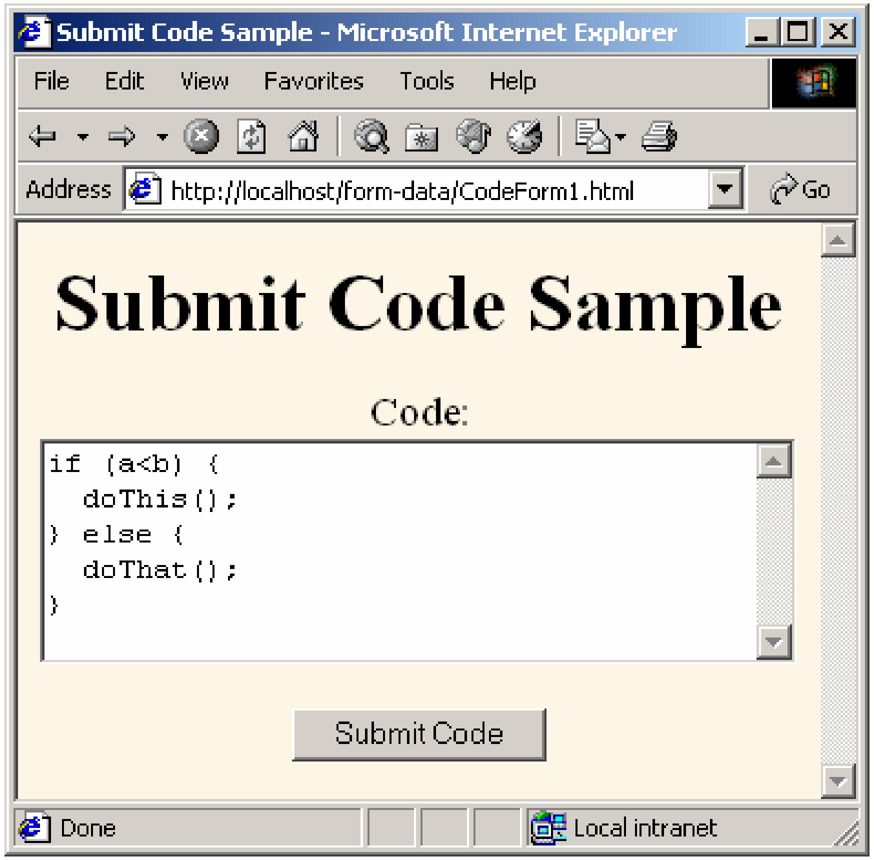

Code Sample - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/servlet/coreservlets.BadCodeServ
Go

Code Sample

```
if (a==b) {
  doThis();
} else {
  doThat();
}
```

Now, wasn't that an interesting sample of code?

A Servlet that Displays Code Samples: Special Chars

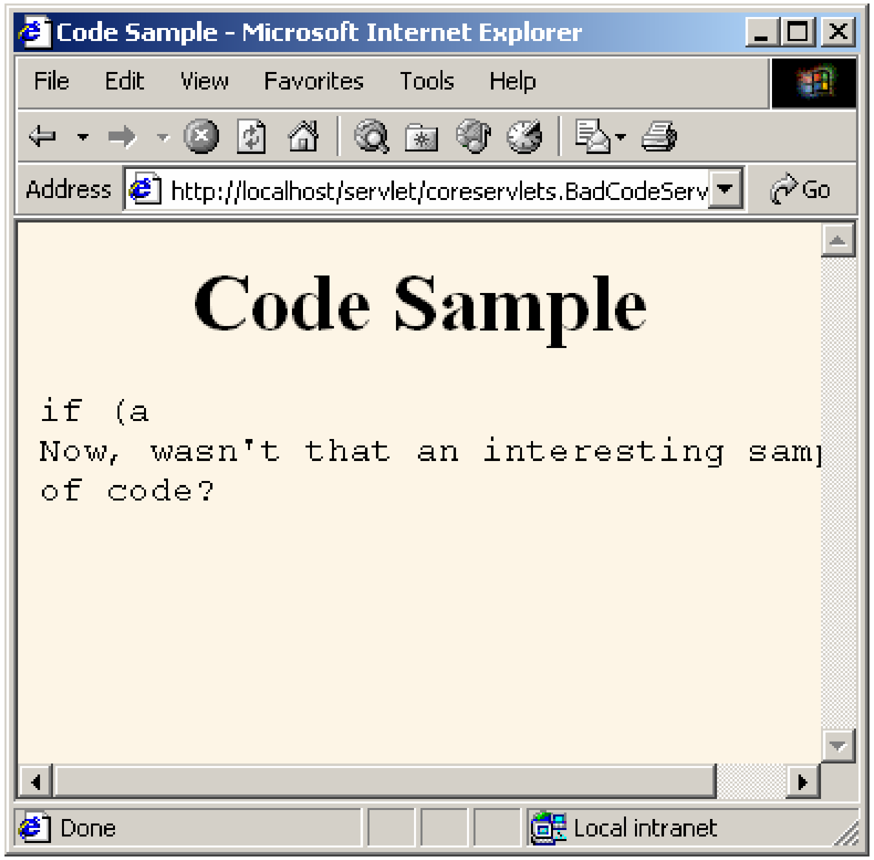

Submit Code Sample - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/form-data/CodeForm1.html
Go

Submit Code Sample

Code:

```
if (a<b) {
  doThis();
} else {
  doThat();
}
```

Submit Code


Code Sample - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/servlet/coreservlets.BadCodeServ
Go

Code Sample

```
if (a
Now, wasn't that an interesting sam
of code?
```

Filtering Strings for HTML-Specific Characters (Code)

```
public class ServletUtilities {
  public static String filter(String input) {
    if (!hasSpecialChars(input)) {
      return(input);
    }
    StringBuffer filtered =
      new StringBuffer(input.length());
    char c;
    for(int i=0; i<input.length(); i++) {
      c = input.charAt(i);
      switch(c) {
        case '<': filtered.append("&lt;"); break;
        case '>': filtered.append("&gt;"); break;
        case '"': filtered.append("&quot;"); break;
        case '&': filtered.append("&amp;"); break;
        default: filtered.append(c);
      }
    }
    return(filtered.toString());
  }
```

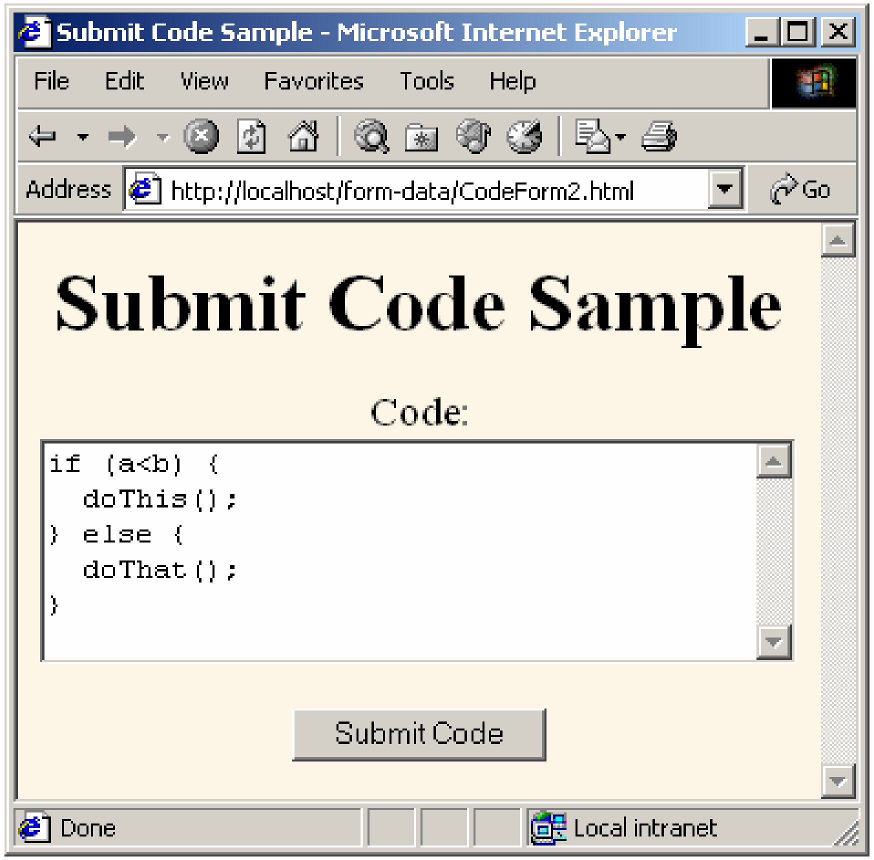
Filtering Strings for HTML-Specific Characters (Code)

```
public static boolean hasSpecialChars(String input) {
    boolean flag = false;
    if ((input !=null) && (input.length()>0)) {
     char c;
     for(int i=0; i<input.length(); i++) {
      c = input.charAt(i);
      switch(c) {
        case '<': flag = true; break;
        case '>': flag = true; break;
        case '"': flag = true; break;
        case '&': flag = true; break;
      } // switch
     } //for loop
    } // if statement
    return(flag);
   } // method hasSpecialChars
  } // class filter
```

A Servlet that Displays Code Samples: Filtering

```
public class GoodCodeServlet extends BadCodeServlet {
  protected String getCode(HttpServletRequest request) {
    return
      (ServletUtilities.filter(super.getCode(request)));
  }
}
```

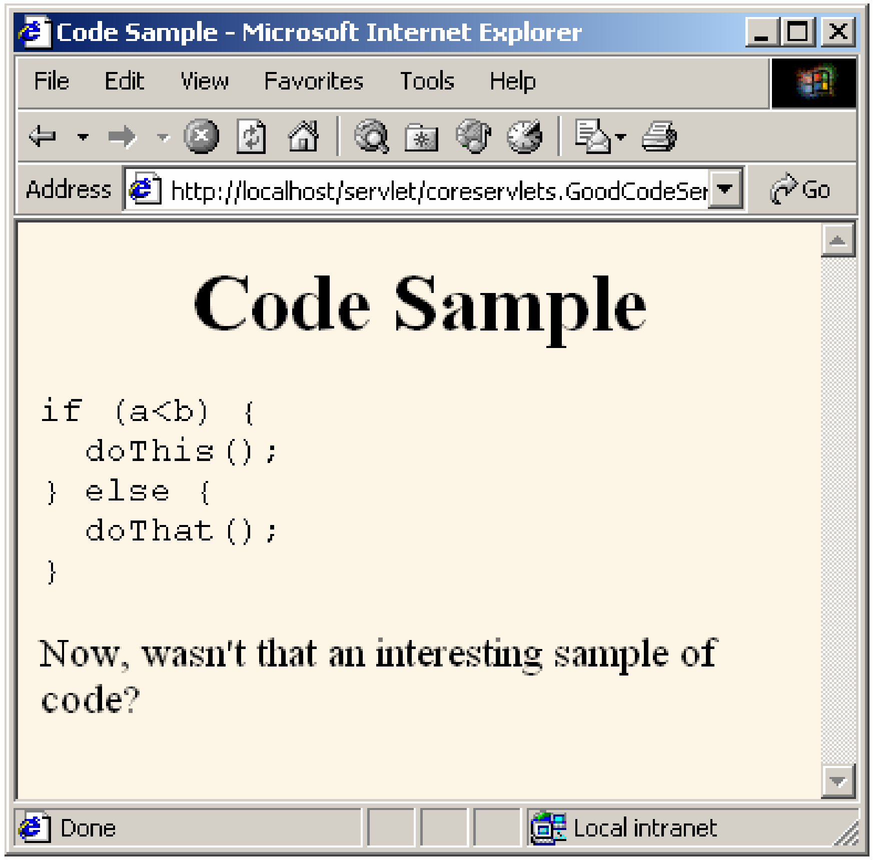
Fixed Servlet that Displays Code Samples: Special Chars


Submit Code Sample - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/form-data/CodeForm2.html
Go
Done
Local intranet

Submit Code Sample

Code:

```
if (a<b) {
  doThis();
} else {
  doThat();
}
```


Code Sample - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost/servlet/coreservlets.GoodCodeSer
Go
Done
Local intranet

Code Sample

```
if (a<b) {
  doThis();
} else {
  doThat();
}
```

Now, wasn't that an interesting sample of code?

Redisplaying the input form when parameters are missing or malformed

- **The same servlet presents the form, processes the data, and presents the results.**
  - The servlet first looks for incoming request data: if it finds none, it presents a blank form.
  - If the servlet finds partial request data, it extracts the partial data, puts it back into the form, and marks the other fields as missing.
  - If the servlet finds the full complement of required data, it processes the request and displays the results.
- **Other options will be presented later on in the course**

Summary

- **Query data comes from HTML forms as URL-encoded name/value pairs**
- **Servlets read data by calling request.getParameter("name")**
  - Results in value as entered into form, not necessarily as sent over network. I.e., *not* URL-encoded.
- **Always check for missing or malformed data**
  - Missing: null or empty string
  - Special case: query data that contains special HTML characters
    - Need to be filtered if query data will be placed into resultant HTML page