

CNT 4714: Enterprise Computing Spring 2012

Introduction to JavaServer Pages (JSP) – Part 2

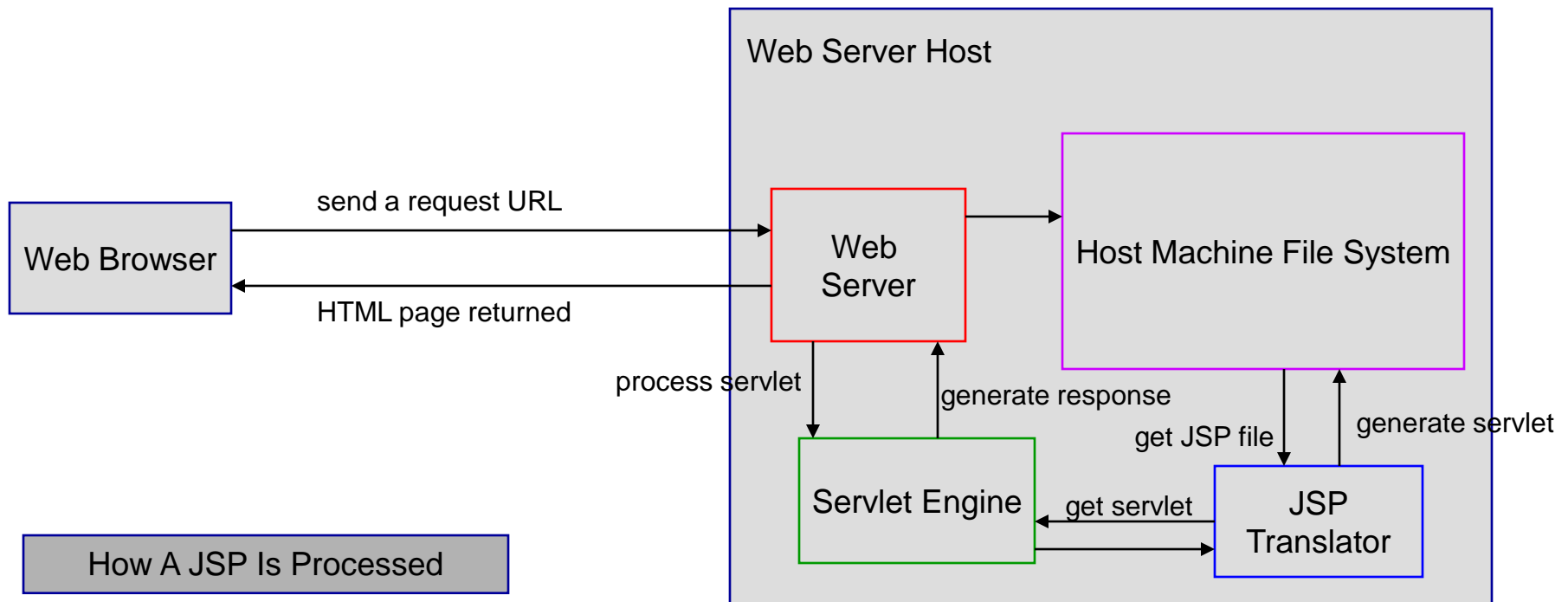
Instructor : Dr. Mark Llewellyn
 markl@cs.ucf.edu
 HEC 236, 407-823-2790
 <http://www.cs.ucf.edu/courses/cnt4714/spr2012>

Department of Electrical Engineering and Computer Science
Computer Science Division
University of Central Florida



How A JSP Is Processed

- Much like a servlet, a JSP must first be processed by a web server before it can be displayed in a web browser. The web server must support JSPs and the JSP page must be stored in a file with a `.jsp` extension. The web server translates the JSP into a Java servlet, compiles the servlet, and executes it. The result of the execution is sent to the browser for display.



More On JSP Scripting Constructs

- There are three main types of JSP constructs: **scripting constructs**, **directives**, and **actions**.
- **Scripting elements** allow you to specify Java code that will become part of the resultant servlet.
- **Directives** enable you to control the overall structure of the resultant servlet.
- **Actions** enable you to control the behavior of the JSP engine.
- We'll look in more detail at all of these, starting with the scripting constructs.



Scripting Constructs

- There are three main types of JSP scripting constructs that can be used to insert Java code into a resultant servlet: **expressions**, **scriptlets** and **declarations**. Recall that there are also **comments** and **escape sequences**.
- A **JSP expression** is used to insert a Java expression directly into the output. It has the following form:

```
<%= java expression %>
```

- The expression is evaluated, converted into a string, and set to the output stream of the servlet.



Scripting Constructs

- A **JSP scriptlet** enables you to insert a Java statement into the servlet's `jspService` method which is invoked by the service method. A JSP scriptlet has the following form:

```
<% java statement %>
```

- A **JSP declaration** is for declaring methods or fields into the servlet. It has the following form:

```
<%! java declaration %>
```

- HTML comments have the form:

```
<!-- HTML comment -->
```

- If you don't want the comment to appear in the resultant HTML file, use a **JSP comment** which has the form:

```
<%-- JSP comment -->
```



*C:\Program Files\Apache Software Foundation\Tomcat 7.0_Tomcat7.0.22\webapps\CNT4714\jsp\ComputeLoan.html - No...

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?



toc.html banner.html ComputeLoan.html

ComputeLoan.html

```

1  <!-- ComputeLoan.html -->
2  <html>
3  <head>
4      <title>ComputeLoan</title>
5  </head>
6  <body bgcolor=white background=images/background.jpg lang=EN-US link=blue vlink=blue
7      style='tab-interval:.5in'>
8  <font size = 5><b>Compute Loan Payment</b></font>
9  <form method="get" action="/CNT4714/jsp/ComputeLoan.jsp"><p><p><br>
10     Loan Amount   <input type="text" name="loanAmount"><br><br>
11     Annual Interest Rate <input type="text" name="annualInterestRate"><br><br>
12     Number of Years <input type="text" name="numberOfYears" size="3"><br></p><p>
13     <input type="submit" name="Submit" value="Compute Loan Payment">
14     <input type="reset" value="Reset">
15     </p>
16 </form>
17 </body>
18 </html>
    
```

Hyper Text M length : 710 lines : 18 Ln : 18 Col : 1 Sel : 0 Macintosh ANSI INS

Start Processing a forward... jsp *C:\Program Files... 4:12 PM

To direct input to this virtual machine, press Ctrl+G.





ComputeLoan.jsp

```

2 <html>
3 <head>
4     <title>ComputeLoan</title>
5 </head><body bgcolor=white background=images/background.jpg lang=EN-US link=blue vlink=blue
6 style='tab-interval:.5in'>
7 <% double loanAmount = Double.parseDouble( request.getParameter("loanAmount"));
8     double annualInterestRate = Double.parseDouble(request.getParameter("annualInterestRate"));
9     double numberOfYears = Integer.parseInt(request.getParameter("numberOfYears"));
10    double monthlyInterestRate = annualInterestRate / 1200;
11    double monthlyPayment = loanAmount * monthlyInterestRate /
12        (1 - 1 / Math.pow(1 + monthlyInterestRate, numberOfYears * 12));
13    double totalPayment = monthlyPayment * numberOfYears * 12;
14 <%>
15 <b><font size = 7> Loan Details </font></b><br><br>
16 <font size = 5>
17 Loan Amount:
18     <%= loanAmount %> <br><br>
19 Annual Interest Rate:
20     <%= annualInterestRate %> <br><br>
21 Number of Years:
22     <%= numberOfYears %> <br><br>
23 <b>
24 Monthly Payment:
25 <%= "$" + monthlyPayment %> <br><br>
26 Total Payment:
27 <%= "$" + totalPayment %> <br><br>
28 </b>
29 </body>

```

JavaServer length : 1089 lines : 30

Ln : 23 Col : 4 Sel : 0

Dos\Windows

ANSI

INS



Opera ComputeLoan

localhost:8080/CNT4714/jsp/ComputeLoan.html

Compute Loan Payment

Loan Amount

Annual Interest Rate

Number of Years

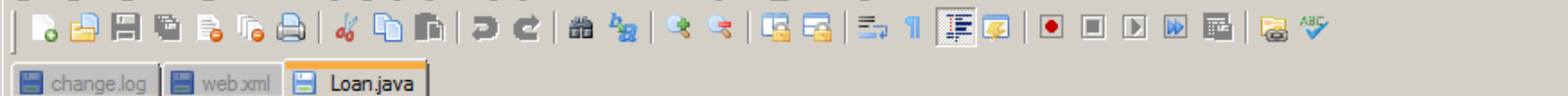


The screenshot shows a web browser window titled "ComputeLoan - Opera". The address bar displays "localhost:8080/CNT4714/jsp/ComputeLoan.jsp". The page content is displayed on a light beige, textured background and includes the following information:

- Loan Amount:** 180000.0
- Annual Interest Rate:** 6.0
- Number of Years:** 15.0
- Monthly Payment:** \$1518.9422904872322
- Total Payment:** \$273409.6122877018

The browser interface includes navigation buttons, a search bar with "Search with Google", and a status bar at the bottom showing "View (120%)". The VMware Player interface is visible at the bottom, including the Start button, taskbar with "ComputeLoan - O...", and system tray with the time "4:20 PM". A message at the bottom left reads "To direct input to this virtual machine, press Ctrl+G."





```
1 package code;
2
3 public class Loan {
4     private double annualInterestRate;
5     private int numOfYears;
6     private double loanAmount;
7     private java.util.Date loanDate;
8
9     /** Default constructor */
10    public Loan() {
11        this(7.5, 30, 100000);
12    }
13
14    /** Construct a loan with specified annual interest rate,
15     *   number of years and loan amount
16     */
17    public Loan(double annualInterestRate, int numOfYears,
18               double loanAmount) {
19        this.annualInterestRate = annualInterestRate;
20        this.numOfYears = numOfYears;
21        this.loanAmount = loanAmount;
22        loanDate = new java.util.Date();
23    }
24
25    /** Return annualInterestRate */
26    public double getAnnualInterestRate() {
27        return annualInterestRate;
28    }
```





change.log web.xml Loan.java

```
29
30  /** Set a new annualInterestRate */
31  public void setAnnualInterestRate(double annualInterestRate) {
32      this.annualInterestRate = annualInterestRate;
33  }
34
35  /** Return numOfYears */
36  public int getNumOfYears() {
37      return numOfYears;
38  }
39
40  /** Set a new numOfYears */
41  public void setNumOfYears(int numOfYears) {
42      this.numOfYears = numOfYears;
43  }
44
45  /** Return loanAmount */
46  public double getLoanAmount() {
47      return loanAmount;
48  }
49
50  /** Set a new loanAmount */
51  public void setLoanAmount(double loanAmount) {
52      this.loanAmount = loanAmount;
53  }
54
55  /** Find monthly payment */
56  public double monthlyPayment() {
```

Java source file length : 1721 lines : 72 Ln : 1 Col : 1 Sel : 0 UNIX ANSI INS





change.log web.xml Loan.java

```
53     }
54
55     /** Find monthly payment */
56     public double monthlyPayment() {
57         double monthlyInterestRate = annualInterestRate / 1200;
58         return loanAmount * monthlyInterestRate / (1 -
59             (Math.pow(1 / (1 + monthlyInterestRate), numOfYears * 12)));
60     }
61
62     /** Find total payment */
63     public double totalPayment() {
64         return monthlyPayment() * numOfYears * 12;
65     }
66
67     /** Return loan date */
68     public java.util.Date getLoanDate() {
69         return loanDate;
70     }
71 }
72
```

Java source file

length : 1721 lines : 72

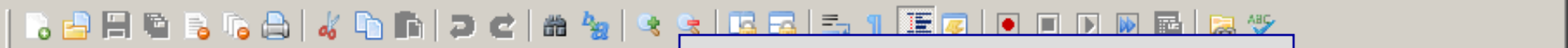
Ln : 1 Col : 1 Sel : 0

UNIX

ANSI

INS





JSP directive to include a Java class.

```

1 <!-- ComputeLoan2.jsp -->
2 <html>
3 <head>
4 <title>ComputeLoan</title>
5 </head><body bgcolor=white background=images/background.jpg lang=EN-US link=blue vlink=blue
6 style='tab-interval:.5in'>
7 <%@ page import = "code.Loan" %>
8
9 <% double loanAmount = Double.parseDouble( request.getParameter("loanAmount"));
10 double annualInterestRate = Double.parseDouble(request.getParameter("annualInterestRate"));
11 int numberOfYears = Integer.parseInt(request.getParameter("numberOfYears"));
12
13 Loan loan = new Loan (annualInterestRate, numberOfYears, loanAmount);
14 %>
15
16 <b><font size = 7> Loan Details </b></font><br><br>
17 <font size = 5>
18 Loan Amount:
19 <%= loanAmount %> <br><br>
20 Annual Interest Rate:
21 <%= annualInterestRate %> <br><br>
22 Number of Years:
23 <%= numberOfYears %> <br><br>
24 <b>
25 Monthly Payment:
26 <%= "$" + loan.monthlyPayment() %> <br><br>
27 Total Payment:
28 <%= "$" + loan.totalPayment() %> <br><br>

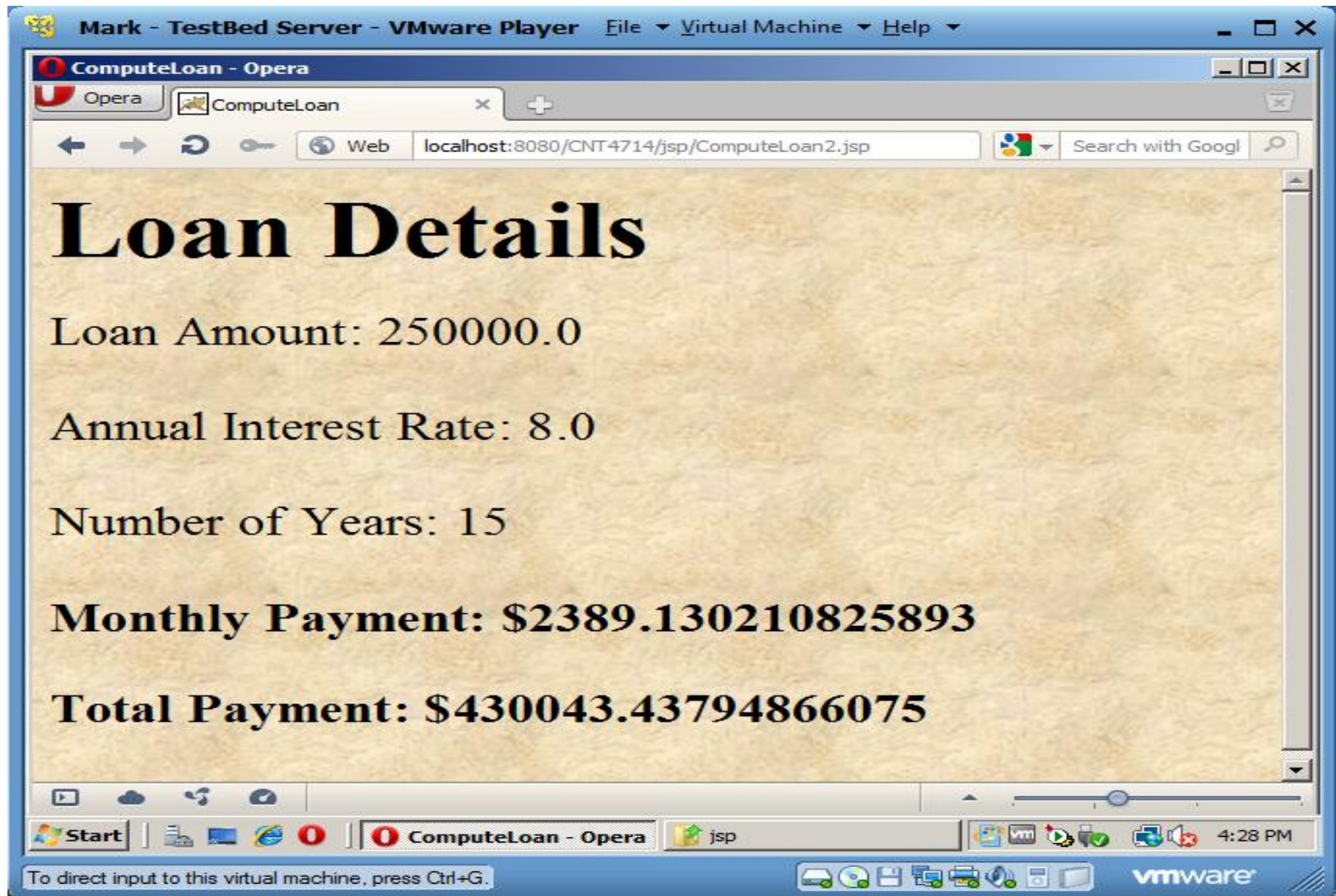
```



Scripting Example Using Directives



Scripting Example Using Directives



The screenshot shows a VMware Player window titled "Mark - TestBed Server - VMware Player". Inside the player, an Opera browser window titled "ComputeLoan - Opera" is open. The browser's address bar shows the URL "localhost:8080/CNT4714/jsp/ComputeLoan2.jsp". The main content of the browser displays the following information:

Loan Details

Loan Amount: 250000.0

Annual Interest Rate: 8.0

Number of Years: 15

Monthly Payment: \$2389.130210825893

Total Payment: \$430043.43794866075

The VMware Player window also shows a taskbar at the bottom with the Start button, several application icons, and a system tray with the time "4:28 PM" and the VMware logo.



JSP Standard Actions

- JSP standard actions provide programmers with access to several of the most common tasks performed in a JSP, such as including content from other resources, forwarding requests to other resources and interacting with JavaBean software components.
- JSP containers process actions at request time.
- Actions are delimited by `<jsp: action>` and `</jsp: action>`, where *action* is the standard action name.
 - In cases where nothing appears between the starting and ending tags, the XML empty element syntax `<jsp: action />` can be used.



JSP Standard Actions

<code><jsp: include></code>	Dynamically includes another resource in a JSP. As the JSP executes, the referenced resource is included and processed.
<code><jsp: forward></code>	Forwards request processing to another JSP, servlet or static page. This action terminates the current JSP's execution.
<code><jsp: plugin></code>	Allows a plug-in component to be added to a page in the form of a browser-specific object or embed HTML element. In the case of a Java applet, this action enables the browser to download and install the Java Plug-in, if it is not already installed on the client computer.
<code><jsp: param></code>	Used with the include, forward and plug-in actions to specify additional name-value pairs of information for use by these actions.



JSP Standard Actions

JavaBean Manipulation	
<code><jsp:useBean></code>	Specifies that the JSP uses a JavaBean instance (i.e., an object of the class that declares the JavaBean). This action specifies the scope of the object and assigns it an ID (i.e., a variable name) that scripting components can use to manipulate the bean.
<code><jsp:setProperty></code>	Sets a property in the specified JavaBean instance. A special feature of this action is automatic matching of request parameters to bean properties of the same name.
<code><jsp:getProperty></code>	Gets a property in the specified JavaBean instance and converts the result to a string for output in the response.



<jsp: include> Action

- JSPs support two include mechanisms – the `<jsp: include>` action and the `include` directive.
- Action `<jsp: include>` enables dynamic content to be included in a JSP at request time. If the included resource changes between requests, the next request to the JSP containing the `<jsp: include>` action includes the resource's new content.
- The `include` directive copies the content into the JSP once, at JSP translation time. If the included resource changes, the new content will not be reflected in the JSP that uses the `include` directive, unless the JSP is recompiled, which would normally occur only if a new version of the JSP were installed.



A JSP Using the `<jsp: include>` Action

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- include.jsp -->

<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title>Using jsp:include</title>
    <style type = "text/css">
      body {
        font-family: tahoma, helvetica, arial, sans-serif;
      }
      table, tr, td {
        font-size: 1.1em;
        border: 3px groove;
        padding: 5px;
        background-color: #dddddd;
      }
    </style>
  </head>
```




```
<body>
  <table>
    <tr>
      <td style = "width: 250px; text-align: center">
        <img src = "smallucf.gif"
          width = "140" height = "93"
          alt = "pegasus logo" />
      </td>
      <td>
        <!-- include banner.html in this JSP -->
        <jsp:include page = "banner.html"
          flush = "true" />
      </td>
    </tr>
    <tr>
      <td style = "width: 250px">
        <!-- include toc.html in this JSP -->
        <jsp:include page = "toc.html" flush = "true" />
      </td>
      <td style = "vertical-align: top">
        <!-- include clock2.jsp in this JSP -->
        <jsp:include page = "clock2.jsp"
          flush = "true" />
      </td>
    </tr>
  </table>
</body>
</html>
```



Banner.html

```
<!-- banner.html -->
<!-- banner to include in another document -->
<div style = "width: 800px">
  <p>
    CNT 4714 - Enterprise Computing
    <br />
    Spring 2012 Semester - University of Central Florida
  </p>
  <p>
    <a href = "mailto:markl@cs.ucf.edu">markl@cs.ucf.edu</a>
  </p>
</div>
```



Table of Contents (toc.html)

```
<!-- toc.html -->
<!-- contents to include in another document -->
<p><a href = "http://www.cs.ucf.edu/courses/cnt4714/spr2012">
    CNT 4714 Course Webpage
</a></p>
<p><a href = "http://www.cs.ucf.edu/faculty/markl.html">
    Instructor's Webpage
</a></p>
<p><a href =
"http://www.cs.ucf.edu/courses/cnt4714/spr2012/code.html">
    Code Download Page
</a></p>
<p><a href =
"http://www.cs.ucf.edu/courses/cnt4714/spr2012/homework.html">
    Programming Assignments Page
</a></p>
<p>Send questions or comments about this site to
    <a href = "mailto:markl@cs.ucf.edu">
        markl@cs.ucf.edu
    </a><br />
</p>
```



Clock2.jsp

```
<!-- clock2.jsp -->
<!-- date and time to include in another document via redirection -->
<table>
  <tr>
    <td style = "background-color: black;">
      <p class = "big" style = "color: cyan; font-size: 3em;
        font-weight: bold;">
        <%-- script to determine client local and --%>
        <%-- format date accordingly --%>
        <%
          // get client locale
          java.util.Locale locale = request.getLocale();

          // get DateFormat for client's Locale
          java.text.DateFormat dateFormat =
            java.text.DateFormat.getDateTimeInstance(
              java.text.DateFormat.LONG,
              java.text.DateFormat.LONG, locale );
        %> <%-- end script --%>
        <%-- output date --%>
        <%= dateFormat.format( new java.util.Date() ) %>
      </p>
    </td>
  </tr>
</table>
```



The screenshot shows a web browser window titled "Using jsp:include - Opera". The address bar shows "localhost:8080/CNT4714/jsp/include.jsp". The page content includes the UCF logo, course information "CNT 4714 - Enterprise Computing Fall 2011 Semester - University of Central Florida", an email address "markl@cs.ucf.edu", and a list of links: "CNT 4714 Course Webpage", "Instructor's Webpage", "Code Download Page", and "Programming Assignments Page". A central black box displays the timestamp "March 1, 2012 10:59:54 AM EST" in cyan text. A blue arrow points from a box labeled "Execution of include.jsp" to the timestamp. The browser's status bar shows "Zoom (80%)". The Windows taskbar at the bottom displays the Start button, several open applications including "Using jsp:include ...", "CNT 4714 - Spring ...", and "jsp", and the system tray with the time "10:59 AM".



<jsp: forward> Action

- JSP action <jsp: forward> enables a JSP to forward request processing to a different resource, such as an error page.
- Request processing by the original JSP terminates as soon as the JSP forwards the request.
- In the next example, this action is illustrated by forwarding a welcome request to another welcome page. JSP forward1.jsp forwards the request to JSP forward2.jsp. The forwarding action requests a date and time at which the original request was received that is forwarded.



Initial Forward JSP (forward1.jsp)

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!-- forward1.jsp -->
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
    <title>Forward request to another JSP</title>
</head>
<body>
    <% // begin scriptlet
        String name = request.getParameter( "firstName" );
        if ( name != null )
        {
    %> <%-- end scriptlet to insert fixed template data --%>

        <jsp:forward page = "forward2.jsp">
            <jsp:param name = "date"
                value = "<%= new java.util.Date() %>" />
        </jsp:forward>

    <% // continue scriptlet
        } // end if
```



Initial Forward JSP (forward1.jsp) (cont.)

```
else
    {
    %> <%-- end scriptlet to insert fixed template data --%>

        <form action = "forward1.jsp" method = "get">
            <p>Type your first name and press Submit</p>

            <p><input type = "text" name = "firstName" />
                <input type = "submit" value = "Submit" />
            </p>
        </form>

        <% // continue scriptlet

            } // end else

        %> <%-- end scriptlet --%>
    </body>
</html> <!-- end XHTML document -->
```



Forward2 JSP (forward2.jsp)

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!-- forward2.jsp -->
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
    <title>Processing a forwarded request</title>
    <style type = "text/css">
        .big
        {
            font-family: tahoma, helvetica, arial, sans-serif;
            font-weight: bold;
            font-size: 2em;
        }
    </style>
</head>
<body>
    <p class = "big">
        Hello <%= request.getParameter( "firstName" ) %>, <br />
        Your redirection request was received <br /> and
        forwarded at
```



Forward2 JSP (forward2.jsp) (cont.)

```
</p>
  <table style = "border: 6px outset;">
    <tr>
      <td style = "background-color: black;">
        <p class = "big" style = "color: cyan;">
          <%= request.getParameter( "date" ) %>
        </p>
      </td>
    </tr>
  </table>
</body>
</html>
```



Forward request to another JSP - Opera

Opera Forward request to anot... x New tab (Ctrl+T)

localhost:8080/CNT4714/jsp/forward1.jsp

Type your first name and press Submit

Mark Submit

Original request is invoked by forward1.jsp

New tab (Ctrl+T)

Start Forward request ... CNT 4714 - Spring ... jsp C:\Program Files\A... 11:00 AM

To direct input to this virtual machine, press Ctrl+G.

vmware



Processing a forwarded request - Opera

Processing a forwarded r... x

Web localhost:8080/CNT4714/jsp/forward1.jsp

Search with Google

Hello Mark, Your redirection request was received and forwarded at

Thu Mar 01 11:01:11 EST 2012

Zoom (120%)

Forward2.jsp receives forwarded request for service with firstName passed as a parameter

