

Chapter 9

How to use JSTL (JSP Standard Tag Library)

Objectives

Applied

1. Use the JSTL core library with EL to add functionality to JSPs.
2. Use the documentation for the JSTL libraries to learn about other JSTL capabilities.

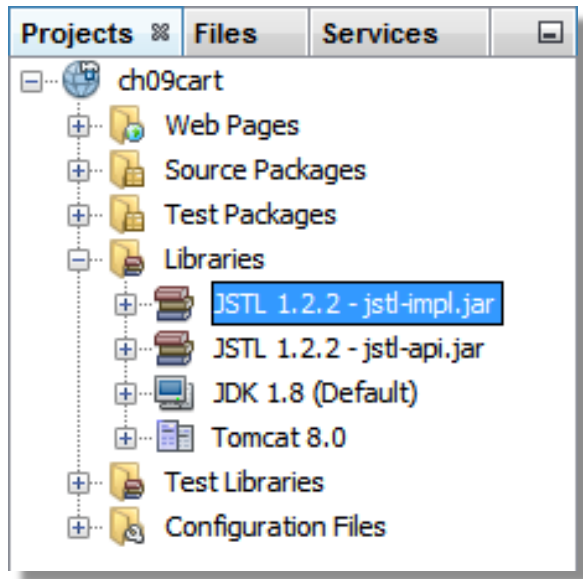
Knowledge

1. Describe the use of these JSTL tags: out, forEach, forToken, if, choose, import, and url.

The primary JSTL libraries

Name	Prefix	URI	Description
Core	c	http://java.sun.com/jsp/jstl/core	Core tags for common tasks such as looping and if/else statements.
Formatting	fmt	http://java.sun.com/jsp/jstl/fmt	Tags for formatting numbers, times, and dates so they work correctly with internationalization (i18n).
SQL	sql	http://java.sun.com/jsp/jstl/sql	Tags for working with SQL queries and data sources.
XML	x	http://java.sun.com/jsp/jstl/xml	Tags for manipulating XML documents.
Functions	fn	http://java.sun.com/jsp/jstl/functions	Functions that can be used to manipulate strings.

NetBeans with the JSTL 1.2 library added



A taglib directive for the JSTL core library

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

A JSTL out tag

```
<td><c:out value="{item.product.description}" /></td>
```

An introduction to JSTL

- The *JSP Standard Tag Library* (*JSTL*) provides tags for common JSP tasks.
- Before you can use JSTL tags within an application, you must make the `jstl-impl.jar` and `jstl-api.jar` files available to the application. To do that with NetBeans, you can add the JSTL 1.2 class library to your project.
- Before you can use JSTL tags within a JSP, you must code a `taglib` directive that identifies the JSTL library and its prefix.

Browser that displays the JSTL documentation

The screenshot shows a web browser window with the address bar containing `docs.oracle.com/javaee/5/jstl/1.1/docs/tlddocs/`. The page title is "out (TLDDoc Generated D...". The main content area displays the documentation for the `<out>` tag, including a navigation menu, a description, and tables for tag information and attributes.

TLDDoc Generated
[All Tags / Functions](#)

Tag Libraries
[JSTL core](#)
[JSTL fmt](#)
[JSTL sql](#)
[JSTL XML](#)
[JSTL functions](#)

JSTL core
Tags
[c:catch](#)
[c:choose](#)
[c:if](#)
[c:import](#)
[c:forEach](#)
[c:forTokens](#)
[c:out](#)
[c:otherwise](#)
[c:param](#)
[c:redirect](#)
[c:remove](#)
[c:set](#)
[c:url](#)

Overview Library Tag Help
[FRAMES](#) [NO FRAMES](#)

JSTL core
Tag out

Like `<%= ... >`, but for expressions.

Tag Information

Tag Class	org.apache.taglibs.standard.tag.rt.core.OutTag
TagExtraInfo Class	None
Body Content	JSP
Display Name	None

Attributes

Name	Required	Request-time	Type	Description
value	true	true	java.lang.String	Expression to be evaluated.
default	false	true	java.lang.String	Default value if the resulting value is null.

XSS attacks

- In a *cross-site scripting (XSS) attack*, an attacker attempts to inject Javascript into your page to trick your users into sending them personal data.
- To prevent an XSS attack, use the `out` tag to escape the output for your application. This tag replaces characters such as the left angle bracket (<) and right angle bracket (>) with their corresponding HTML entities.
- It's generally considered a best practice to escape all output that's sent to the browser, including other parts of the HTTP request such as cookies and other headers.

How to escape output entered by the user

```
<label>Email:</label>  
<span><c:out value="\${user.email}" /></span><br>  
<label>First Name:</label>  
<span><c:out value="\${user.firstName}" /></span><br>  
<label>Last Name:</label>  
<span><c:out value="\${user.lastName}" /></span><br>
```

Output for an XSS attack

Thanks for joining our email list

Here is the information that you entered:

Email: jsmith@gmail.com
First Name: John
Last Name: <script>This JavaScript is displayed, not
 executed</script>

How to escape output that displays cookies

```
<table>
  <tr>
    <th>Name</th>
    <th>Value</th>
  </tr>
  <c:forEach var="cook" items="{cookie}">
    <tr>
      <td><c:out value="{cook.value.name}" /></td>
      <td><c:out value="{cook.value.value}" /></td>
    </tr>
  </c:forEach>
</table>
```

How to display a default value

Using the default attribute

```
<p><c:out value="{message}" default="No message" /></p>
```

Using the tag's body

```
<p><c:out value="{message}">No message</c:out></p>
```

How to use single quotes within double quotes

```
<input type="hidden" name="productCode"  
value="<c:out value='{item.product.code}' />">
```

How to loop through a collection

```
<c:forEach var="item" items="${cart.items}">
<tr>
  <td>${item.quantity}</td>
  <td>${item.product.description}</td>
  <td>${item.product.priceCurrencyFormat}</td>
  <td>${item.totalCurrencyFormat}</td>
</tr>
</c:forEach>
```

A cart with two items

Your cart

Quantity	Description	Price	Amount
1	86 (the band) - True Life Songs and Pictures	\$14.95	\$14.95
1	Puddlefoot - The first CD	\$12.95	\$12.95

To change the quantity, enter the new quantity and click on the Update button.

Continue Shopping

Checkout

How to loop through a comma-delimited string

Servlet code

```
session.setAttribute("productCodes", "8601,pf01,pf02,jr01");
```

JSP code

```
<p>Product codes</p>
```

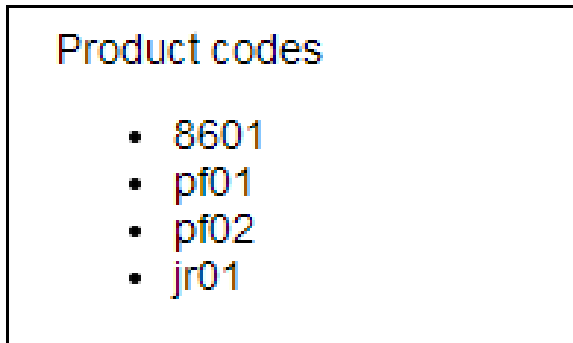
```
<ul>
```

```
<c:forEach var="productCode" items="{productCodes}" delims="," >  
    <li>${productCode}</li>
```

```
</c:forEach>
```

```
</ul>
```

Result displayed in the browser



How to parse a string

Servlet code

```
session.setAttribute("emailAddress", "jsmith@gmail.com");
```

JSP code

```
<p>Email parts</p>
```

```
<ul>
```

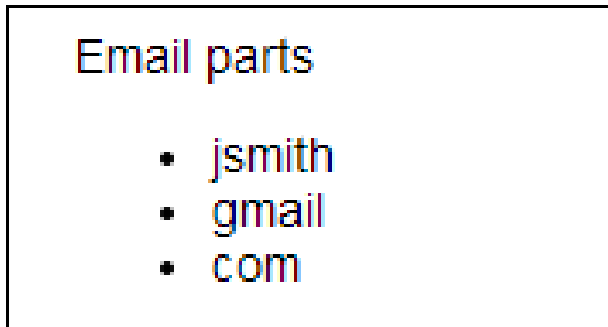
```
<c:forTokens var="part" items="{emailAddress}" delims="@.">
```

```
<li>{part}</li>
```

```
</c:forTokens>
```

```
</ul>
```

Result displayed in the browser



Attributes for advanced loops

Attribute	Description
begin	Specifies the first index for the loop.
end	Specifies the last index for the loop.
step	Specifies the amount to increment the index each time through the loop.
varStatus	Specifies the name of a variable that can be used to get information about the status of the loop. This variable provides the first, last, index, and count properties.

Note

- If necessary, you can nest one loop within another.

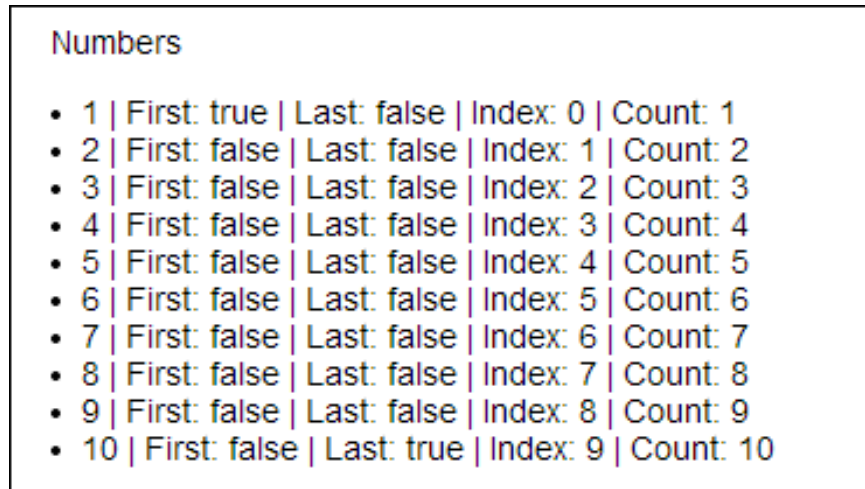
Servlet code that creates an array of numbers

```
int[] numbers = new int[30];
for (int i = 0; i < 30; i++) {
    numbers[i] = i+1;
}
session.setAttribute("numbers", numbers);
```

JSP code that uses advanced attributes

```
<p>Numbers</p>
<ul>
  <c:forEach items="{numbers}" var="number"
            begin="0" end="9" step="1"
            varStatus="status">
    <li>${number} | First: ${status.first} | Last: ${status.last} |
      Index: ${status.index} | Count: ${status.count} </li>
  </c:forEach>
</ul>
```

Result displayed in the browser



Numbers

- 1 | First: true | Last: false | Index: 0 | Count: 1
- 2 | First: false | Last: false | Index: 1 | Count: 2
- 3 | First: false | Last: false | Index: 2 | Count: 3
- 4 | First: false | Last: false | Index: 3 | Count: 4
- 5 | First: false | Last: false | Index: 4 | Count: 5
- 6 | First: false | Last: false | Index: 5 | Count: 6
- 7 | First: false | Last: false | Index: 6 | Count: 7
- 8 | First: false | Last: false | Index: 7 | Count: 8
- 9 | First: false | Last: false | Index: 8 | Count: 9
- 10 | First: false | Last: true | Index: 9 | Count: 10

How to code an if statement

```
<c:if test="\${cart.count == 1}">
    <p>You have 1 item in your cart.</p>
</c:if>
<c:if test="\${cart.count > 1}">
    <p>You have \${cart.count} items in your cart.</p>
</c:if>
```

A cart with two items

Your cart

Quantity	Description	Price	Amount	
<input type="text" value="1"/> <input type="button" value="Update"/>	86 (the band) - True Life Songs and Pictures	\$14.95	\$14.95	<input type="button" value="Remove Item"/>
<input type="text" value="1"/> <input type="button" value="Update"/>	Paddlefoot - The first CD	\$12.95	\$12.95	<input type="button" value="Remove Item"/>

You have 2 items in your cart.

To change the quantity, enter the new quantity and click on the Update button.

How to code an if/else statement

```
<c:choose>
  <c:when test="{cart.count == 0}">
    <p>Your cart is empty.</p>
  </c:when>
  <c:when test="{cart.count == 1}">
    <p>You have 1 item in your cart.</p>
  </c:when>
  <c:otherwise>
    <p>You have ${cart.count} items in your cart.</p>
  </c:otherwise>
</c:choose>
```

A cart that has two items

Your cart

Quantity	Description	Price	Amount	
<input type="text" value="1"/> <input type="button" value="Update"/>	86 (the band) - True Life Songs and Pictures	\$14.95	\$14.95	<input type="button" value="Remove Item"/>
<input type="text" value="1"/> <input type="button" value="Update"/>	Puddlefoot - The first CD	\$12.95	\$12.95	<input type="button" value="Remove Item"/>

You have 2 items in your cart.

To change the quantity, enter the new quantity and click on the Update button.

The if and choose tags

- Use the if tag to perform conditional processing that's similar to an if statement in Java.
- Use the choose tag to perform conditional processing similar to an if/else statement in Java. To do that, code multiple when tags and a single otherwise tag within the choose tag.
- If necessary, nest one if or choose tag within another.

A link that doesn't use the url tag

```
<a href="cart">Continue Shopping</a>
```

How to use the url tag

```
<a href="<c:url value='/cart' />">Continue Shopping</a>
```

How to add a parameter

```
<a href="<c:url value='/cart?productCode=8601' />">  
  Add To Cart  
</a>
```

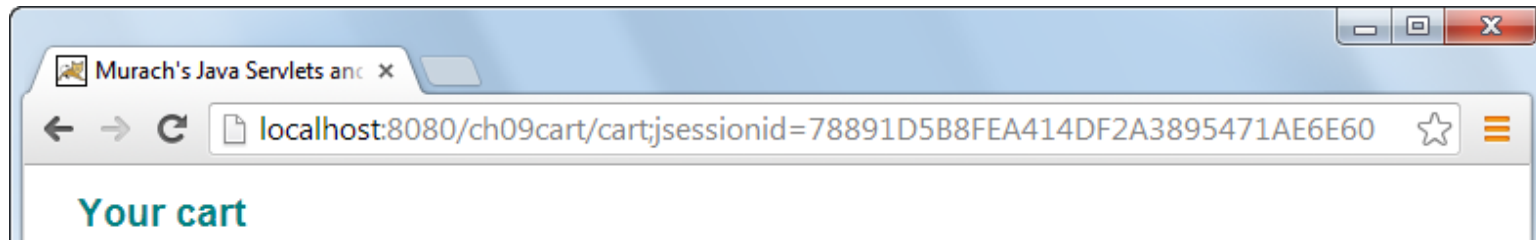
How to use EL to specify a parameter's value

```
<a href="<c:url value='/cart?productCode=${product.code}' />">  
  Add To Cart</a>
```

Another way to code the previous example

```
<a href="<c:url value='/cart'>  
  <c:param name='productCode' value='${product.code}' />  
</c:url>">  
  Add To Cart</a>
```

A browser with the session ID encoded in its URL



A web.xml file fragment that turns off URL encoding

```
<session-config>  
    <session-timeout>30</session-timeout>  
    <tracking-mode>COOKIE</tracking-mode>  
</session-config>
```

The url tag

- You can use the JSTL url tag to encode URLs within your web application that are relative to the application's root directory. As a result, it can prevent your code from breaking if the application context changes.
- By default, the url tag automatically rewrites the URL to include the session ID whenever necessary. This can lead to session hijacking. To prevent this, add a tracking-mode element to the web.xml file to specify the application should only use cookies (not URL encoding) for tracking.
- Use the JSTL param tag if you want to automatically encode unsafe characters such as spaces with special characters such as plus signs.

Other tags in the JSTL core library

Tag name	Description
set	Sets the value of an attribute in a scope.
remove	Removes an attribute from a scope.
catch	Catches any exception that occurs in its body and optionally creates an EL variable that refers to the Throwable object for the exception.
redirect	Redirects the client browser to a new URL.
param	Adds a parameter to the parent tag.

How to set a value in an attribute

```
<c:set var="message" scope="session" value="Test message" />
```

How to set a value in a JavaBean

```
<c:set target="{user}" property="firstName" value="John" />
```

How to remove an attribute

```
<c:remove var="message" scope="session" />
```

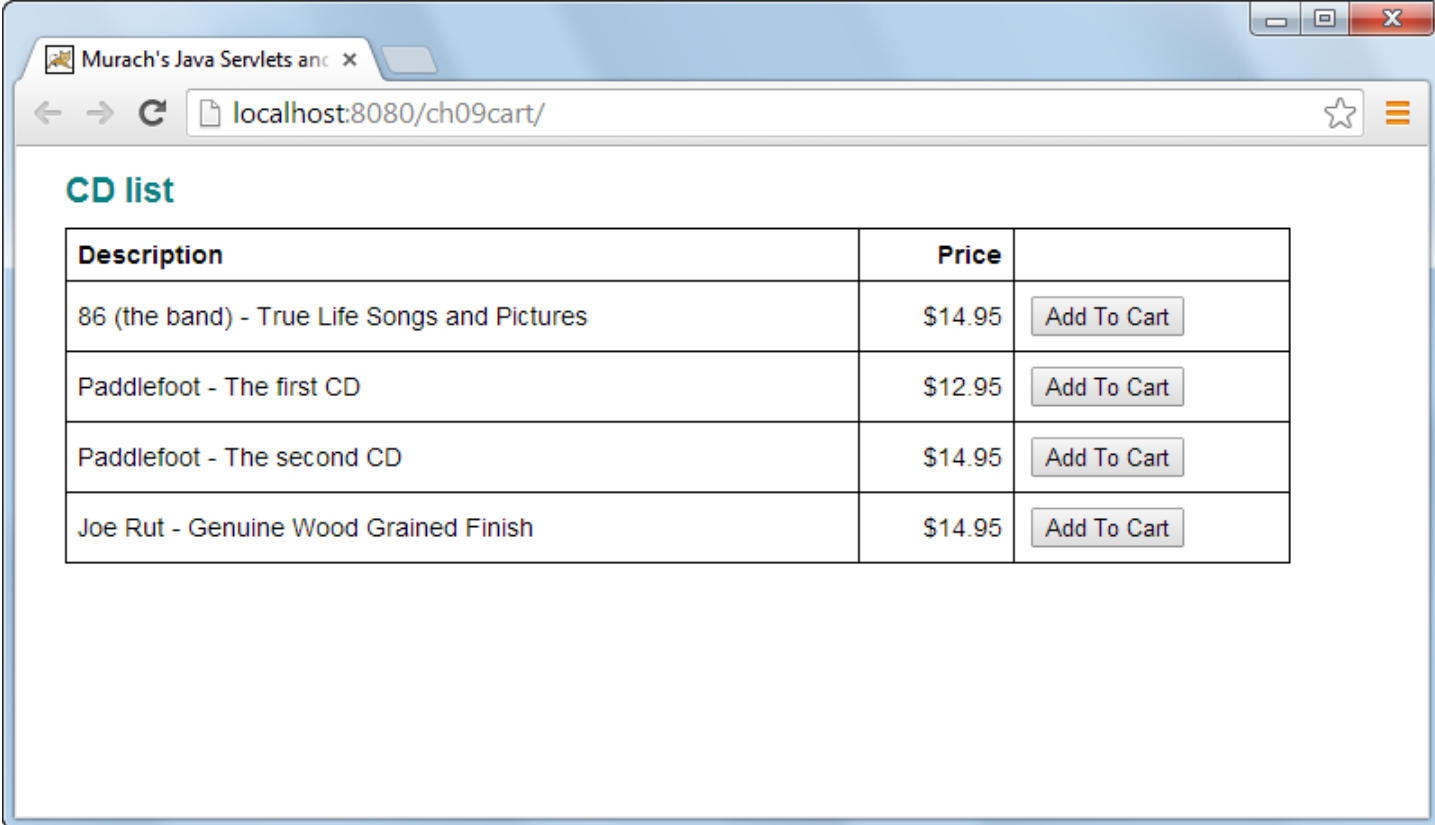

How to catch an exception

```
<c:catch var="e">
  <% // this scriptlet statement throws an exception
      int i = 1/0;
  %>
  <p>Result: <c:out value="{i}" /></p>
</c:catch>
<c:if test="{e != null}">
  <p>An exception occurred. Message: {e.message}</p>
</c:if>
```

How to redirect to another page

```
<c:if test="{e != null}">
  <c:redirect url="/error_java.jsp" />
</c:if>
```

The Index page



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/ch09cart/'. The page content includes a heading 'CD list' and a table with the following data:

Description	Price	
86 (the band) - True Life Songs and Pictures	\$14.95	<input type="button" value="Add To Cart"/>
Paddlefoot - The first CD	\$12.95	<input type="button" value="Add To Cart"/>
Paddlefoot - The second CD	\$14.95	<input type="button" value="Add To Cart"/>
Joe Rut - Genuine Wood Grained Finish	\$14.95	<input type="button" value="Add To Cart"/>

The Cart page

Your cart

Quantity	Description	Price	Amount	
<input type="text" value="1"/> <input type="button" value="Update"/>	86 (the band) - True Life Songs and Pictures	\$14.95	\$14.95	<input type="button" value="Remove Item"/>
<input type="text" value="1"/> <input type="button" value="Update"/>	Paddlefoot - The first CD	\$12.95	\$12.95	<input type="button" value="Remove Item"/>

To change the quantity, enter the new quantity and click on the Update button.

The Product class

```
package murach.business;

import java.io.Serializable;
import java.text.NumberFormat;

public class Product implements Serializable {

    private String code;
    private String description;
    private double price;

    public Product() {
        code = "";
        description = "";
        price = 0;
    }

    public void setCode(String code) {
        this.code = code;
    }

    public String getCode() {
        return code;
    }
}
```

The Product class (continued)

```
public void setDescription(String description) {
    this.description = description;
}

public String getDescription() {
    return description;
}

public void setPrice(double price) {
    this.price = price;
}

public double getPrice() {
    return price;
}

public String getPriceCurrencyFormat() {
    NumberFormat currency = NumberFormat.getCurrencyInstance();
    return currency.format(price);
}
}
```

The LineItem class

```
package murach.business;

import java.io.Serializable;
import java.text.NumberFormat;

public class LineItem implements Serializable {

    private Product product;
    private int quantity;

    public LineItem() {}

    public void setProduct(Product p) {
        product = p;
    }

    public Product getProduct() {
        return product;
    }
}
```

The LineItem class (continued)

```
public void setQuantity(int quantity) {
    this.quantity = quantity;
}

public int getQuantity() {
    return quantity;
}

public double getTotal() {
    double total = product.getPrice() * quantity;
    return total;
}

public String getTotalCurrencyFormat() {
    NumberFormat currency = NumberFormat.getCurrencyInstance();
    return currency.format(this.getTotal());
}
}
```

The Cart class

```
package murach.business;

import java.io.Serializable;
import java.util.ArrayList;

public class Cart implements Serializable {

    private ArrayList<LineItem> items;

    public Cart() {
        items = new ArrayList<LineItem>();
    }

    public ArrayList<LineItem> getItems() {
        return items;
    }

    public int getCount() {
        return items.size();
    }
}
```


The Cart class (continued)

```
public void addItem(LineItem item) {
    String code = item.getProduct().getCode();
    int quantity = item.getQuantity();
    for (int i = 0; i < items.size(); i++) {
        LineItem lineItem = items.get(i);
        if (lineItem.getProduct().getCode().equals(code)) {
            lineItem.setQuantity(quantity);
            return;
        }
    }
    items.add(item);
}
```

```
public void removeItem(LineItem item) {
    String code = item.getProduct().getCode();
    for (int i = 0; i < items.size(); i++) {
        LineItem lineItem = items.get(i);
        if (lineItem.getProduct().getCode().equals(code)) {
            items.remove(i);
            return;
        }
    }
}
```

The index.jsp file

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Murach's Java Servlets and JSP</title>
  <link rel="stylesheet" href="styles/main.css" type="text/css"/>
</head>
<body>

<h1>CD list</h1>
<table>
  <tr>
    <th>Description</th>
    <th class="right">Price</th>
    <th>&nbsp;</th>
  </tr>
  <tr>
    <td>86 (the band) - True Life Songs and Pictures</td>
    <td class="right">$14.95</td>
    <td><form action="cart" method="post">
      <input type="hidden" name="productCode" value="8601">
      <input type="submit" value="Add To Cart">
    </form></td>
  </tr>
```

The index.jsp file (continued)

```
<tr>
  <td>Paddlefoot - The first CD</td>
  <td class="right">$12.95</td>
  <td><form action="cart" method="post">
    <input type="hidden" name="productCode" value="pf01">
    <input type="submit" value="Add To Cart">
  </form></td>
</tr>
<tr>
  <td>Paddlefoot - The second CD</td>
  <td class="right">$14.95</td>
  <td><form action="cart" method="post">
    <input type="hidden" name="productCode" value="pf02">
    <input type="submit" value="Add To Cart">
  </form></td>
</tr>
```

The index.jsp file (continued)

```
<tr>
  <td>Joe Rut - Genuine Wood Grained Finish</td>
  <td class="right">$14.95</td>
  <td><form action="cart" method="post">
    <input type="hidden" name="productCode" value="jr01">
    <input type="submit" value="Add To Cart">
  </form></td>
</tr>
</table>

</body>
</html>
```

The CartServlet class

```
package murach.cart;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

import murach.data.*;
import murach.business.*;

public class CartServlet extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        ServletContext sc = getServletContext();

        // get current action
        String action = request.getParameter("action");
        if (action == null) {
            action = "cart"; // default action
        }
    }
}
```

The CartServlet class (continued)

```
// perform action and set URL to appropriate page
String url = "/index.jsp";
if (action.equals("shop")) {
    url = "/index.jsp";    // the "index" page
}
else if (action.equals("cart")) {
    String productCode = request.getParameter("productCode");
    String quantityString = request.getParameter("quantity");

    HttpSession session = request.getSession();
    Cart cart = (Cart) session.getAttribute("cart");
    if (cart == null) {
        cart = new Cart();
    }
}
```

The CartServlet class (continued)

```
//if the user enters a negative or invalid quantity,  
//the quantity is automatically reset to 1.  
int quantity;  
try {  
    quantity = Integer.parseInt(quantityString);  
    if (quantity < 0) {  
        quantity = 1;  
    }  
} catch (NumberFormatException nfe) {  
    quantity = 1;  
}  
  
String path = sc.getRealPath("/WEB-INF/products.txt");  
Product product = ProductIO.getProduct(productCode, path);  
  
LineItem lineItem = new LineItem();  
lineItem.setProduct(product);  
lineItem.setQuantity(quantity);  
if (quantity > 0) {  
    cart.addItem(lineItem);  
} else if (quantity == 0) {  
    cart.removeItem(lineItem);  
}
```

The CartServlet class (continued)

```
        session.setAttribute("cart", cart);
        url = "/cart.jsp";
    }
    else if (action.equals("checkout")) {
        url = "/checkout.jsp";
    }

    sc.getRequestDispatcher(url)
        .forward(request, response);
    }
}
```

Note

- In the web.xml file, the CartServlet class is mapped to the “/cart” URL.

The cart.jsp file

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Murach's Java Servlets and JSP</title>
  <link rel="stylesheet" href="styles/main.css" type="text/css"/>
</head>
<body>

<h1>Your cart</h1>

<table>
  <tr>
    <th>Quantity</th>
    <th>Description</th>
    <th>Price</th>
    <th>Amount</th>
    <th></th>
  </tr>
```

The cart.jsp file (continued)

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:forEach var="item" items="${cart.items}">
  <tr>
    <td>
      <form action="" method="post">
        <input type="hidden" name="productCode"
          value="<c:out value='${item.product.code}' />">
        <input type="text" name="quantity"
          value="<c:out value='${item.quantity}' />" id="quantity">
        <input type="submit" value="Update">
      </form>
    </td>
    <td><c:out value='${item.product.description}' /></td>
    <td>${item.product.priceCurrencyFormat}</td>
    <td>${item.totalCurrencyFormat}</td>
    <td>
      <form action="" method="post">
        <input type="hidden" name="productCode"
          value="<c:out value='${item.product.code}' />">
        <input type="hidden" name="quantity" value="0">
        <input type="submit" value="Remove Item">
      </form>
    </td>
  </tr>
</c:forEach>
```

The cart.jsp file (continued)

```
</c:forEach>
```

```
</table>
```

```
<p><b>To change the quantity</b>, enter the new quantity  
and click on the Update button.</p>
```

```
<form action="" method="post">  
  <input type="hidden" name="action" value="shop">  
  <input type="submit" value="Continue Shopping">  
</form>
```

```
<form action="" method="post">  
  <input type="hidden" name="action" value="checkout">  
  <input type="submit" value="Checkout">  
</form>
```

```
</body>  
</html>
```