Adv. Java means DURGA SIR..
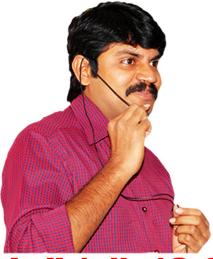
ADV.JAVA
With
SCWCD / OCWCD
Question Bank

Chapter : 8. Building JSP Pages Using Standard Actions

DURGA M.Tech

(Sun certified & Realtime Expert)

Ex. IBM Employee

Trained Lakhs of Students
for last 14 years across INDIA

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

Plot No : 202, IInd Floor ,HUDA Maitrivanam,Ameerpet, Hyd-5000038.
www.durgasoft.com

8. Building JSP Pages Using Standard Actions

- Given a design goal, create a code snippet using the following standard actions: jsp:useBean (with attributes: 'id', 'scope', 'type', and 'class'), jsp:getProperty, jsp:setProperty (with all attribute combinations), and jsp:attribute.

- Given a design goal, create a code snippet using the following standard actions: jsp:include, jsp:forward, and jsp:param

Q1. Given the JSP code:
10. <html>
11. <body>
12. <jsp:useBean id='customer' class='com.example.Customer' />
13. Hello, ${customer.title} ${customer.lastName}, welcome
14. to Squeaky Beans, Inc.
15. </body>
16. </html>

Which three types of JSP code are used? (Choose three.)

A. Java code   B. template text
C. scripting code   D. standard action
E. expression language

Answer: B, D, E

Q2. Which JSP standard action can be used to import content from a resource called foo.jsp?

A. <jsp:import file='foo.jsp' />
B. <jsp:import page='foo.jsp' />
C. <jsp:include page='foo.jsp' />
D. <jsp:include file='foo.jsp' />
E. <jsp:import>foo.jsp</jsp:import>
F. <jsp:include>foo.jsp</jsp:include>

Answer: C

Q3. Click the Task button.
A servlet context listener loads a list of com.example.Product objects from a database and stores that list into the catalog attribute of the ServletContext object.Place code snippets to construct a jsp:useBean standard action to access this catalog.

```
Drag and Drop                                    _ □ X

A servlet context listener loads a list of com.example.Product objects from
a database and stores that list into the catalog attribute of the
ServletContext object.

Place code snippets to construct a jsp:useBean standard action to
access this catalog.

The jsp:useBean Standard Action

<jsp:useBean
              ┌─────────────────────────────┐
              │        Place here.          │
              ├─────────────────────────────┤
              │        Place here.          │
              ├─────────────────────────────┤
              │        Place here.          │
              └─────────────────────────────┘

/>

Code Snippets

┌─────────────────────────┐   ┌─────────────────────────┐
│      id='product'       │   │   scope='application'   │
├─────────────────────────┤   ├─────────────────────────┤
│   type='java.util.List' │   │      id='catalog'       │
├─────────────────────────┤   ├─────────────────────────┤
│      name='catalog'     │   │     scope='context'     │
├─────────────────────────┤   ├─────────────────────────┤
│  scope='servletContext' │   │ type='com.example.Product' │
└─────────────────────────┘   └─────────────────────────┘

                  ┌──────────────┐
                  │     Done     │
                  └──────────────┘
```

Answer:
<jsp:useBean id='catalog' type='java.util.List' scope='application' />

Q4. A session-scoped attribute is stored by a servlet, and then that servlet
forwards to a JSP page. Which three jsp:useBean attributes must be used to access
this attribute in the JSP page? (Choose three.)

A. id   B. name
C. bean   D. type
E. scope   F. beanName

Answer: A, D, E

Q5. You need to create a JavaBean object that is used only within the current JSP page. It must NOT be accessible to any other page including those that this page might import. Which JSP standard action can accomplish this goal?

A. <jsp:useBean id='pageBean' type='com.example.MyBean' />
B. <jsp:useBean id='pageBean' class='com.example.MyBean' />
C. <jsp:makeBean id='pageBean' type='com.example.MyBean' />
D. <jsp:makeBean id='pageBean' class='com.example.MyBean' />
E. <jsp:useBean name='pageBean' class='com.example.MyBean' />
F. <jsp:makeBean name='pageBean' class='com.example.MyBean' />

Answer: B

Q6. Assume a JavaBean com.example.GradedTestBean exists and has two attributes. The attribute name is of type java.lang.String and the attribute score is of type java.lang.Integer.

An array of com.example.GradedTestBean objects is exposed to the page in a request-scoped attribute called results. Additionally, an empty java.util.HashMap called resultMap is placed in the page scope.

A JSP page needs to add the first entry in results to resultMap, storing the name attribute of the bean as the key and the score attribute of the bean as the value.

Which code snippet of JSTL code satisfies this requirement?

A. ${resultMap[results[0].name] = results[0].score}
B. <c:set var="${resultMap}" key="${results[0].name}"
value="${results[0].score}" />
C. <c:set var="resultMap" property="${results[0].name}">
${results[0].value}
</c:set>
D. <c:set var="resultMap" property="${results[0].name}"
value="${results[0].score}" />
E. <c:set target="${resultMap}" property="${results[0].name}"
value="${results[0].score}" />

Answer: E

ADV.JAVA Means DURGA SIR

Q7. A JSP page needs to set the property of a given JavaBean to a value that is calculated with the JSP page. Which three jsp:setProperty attributes must be used to perform this initialization? (Choose three.)

A. id   B. val
C. name   D. param
E. value   F. property
G. attribute

Answer: C, E, F

Q8. Your web application views all have the same header, which includes the <title> tag in the <head> element of the rendered HTML. You have decided to remove this redundant HTML code from your JSPs and put it into a single JSP called /WEB-INF/jsp/header.jsp. However, the title of each page is unique, so you have decided to use a variable called pageTitle to parameterize this in the header JSP, like this:

10. <title>${param.pageTitle}<title>

Which JSP code snippet should you use in your main view JSPs to insert the header and pass the pageTitle variable?

A. <jsp:insert page='/WEB-INF/jsp/header.jsp'>
${pageTitle='Welcome Page'}
</jsp:insert>
B. <jsp:include page='/WEB-INF/jsp/header.jsp'>
${pageTitle='Welcome Page'}
</jsp:include>
C. <jsp:include file='/WEB-INF/jsp/header.jsp'>
${pageTitle='Welcome Page'}
</jsp:include>
D. <jsp:insert page='/WEB-INF/jsp/header.jsp'>
<jsp:param name='pageTitle' value='Welcome Page' />
</jsp:insert>
E. <jsp:include page='/WEB-INF/jsp/header.jsp'>
<jsp:param name='pageTitle' value='Welcome Page' />
</jsp:include>

Answer: E

Q9. A JSP page needs to instantiate a JavaBean to be used by only that page.Which two jsp:useBean attributes must be used to access this attribute in the JSP page? (Choose two.)

A. id   B. type
C. name   D. class
E. scope   F. create

Answer: A, D

Q10. Click the Exhibit button
Given the HTML form:
1. <html>
2. <body>
3. <form action="submit.jsp">
**4. Name: <input type="text" name="i1">
**
**5. Price: <input type="text" name="i2">
**
6. <input type="submit">
7. </form>
8. </body>
9. </html>

Assume the product attribute does NOT yet exist in any scope.
Which code snippet, in submit.jsp, instantiates an instance of com.example.Product
that contains the results of the form submission?

```
 1.  package com.example;
 2.
 3.  public class Product {
 4.     private String name;
 5.     private double price;
 6.
 7.     public Product() {
 8.        this( "Default", 0.0 );
 9.     }
10.
11.     public Product( String name, double
price ) {
12.        this.name = name;
13.        this.price = price;
14.     }
15.
16.     public String getName() {
17.        return name;
18.     }
19.
20.     public void setName(String name) {
21.        this.name = name;
22.     }
23.
24.     public double getPrice() {
25.        return price;
26.     }
27.
28.     public void setPrice(double price) {
29.        this.price = price;
30.     }
31.  }
```
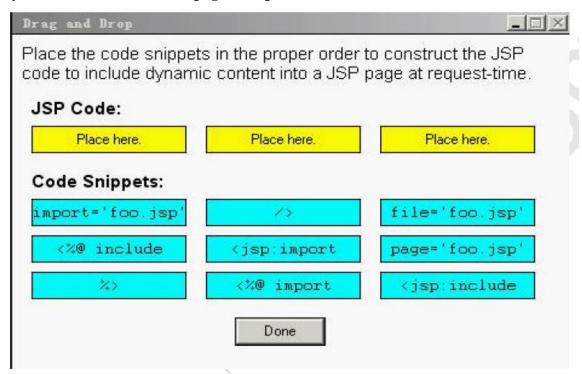
A. <jsp:useBean id="com.example.Product" />
<jsp:setProperty name="product" property="*" />
B. <jsp:useBean id="product" class="com.example.Product" />
${product.name = param.i1}
${product.price = param.i2}
C. <jsp:useBean id="product" class="com.example.Product">
<jsp:setProperty name="product" property="name"
param="i1" />
<jsp:setProperty name="product" property="price"
param="i2" />
</jsp:useBean>
D. <jsp:useBean id="product" type="com.example.Product">
<jsp:setProperty name="product" property="name"
value="<%= request.getParameter("i1") %>" />
<jsp:setProperty name="product" property="price"

Page 7

Plot No : 202, IInd Floor ,HUDA Maitrivanam,Ameerpet, Hyd-5000038.

www.durgasoft.com

value="<%= request.getParameter("i2") %>" />
</jsp:useBean>
Answer: C
Q11. Click the Task button.
Place the code snippets in the proper order to construct the JSP code to include dynamic content into a JSP page at request-time.

Ans:
<jsp:include page='foo.jsp' />

Q12. Click the Exhibit button.
Given:
10. <form action='create_product.jsp'>
**11. Product Name: <input type='text' name='prodName'/>
**
**12. Product Price: <input type='text' name='prodPrice'/>
**
13. </form>
For a given product instance, which three jsp:setProperty attributes must be used to initialize its properties from the HTML form? (Choose three.)

```
1.  package com.example;
2.
3.  public class Product {
4.     private String name;
5.     private double price;
6.
7.     public Product() {
8.        this( "Default", 0.0 );
9.     }
10.
11.    public Product( String name, double
price ) {
12.       this.name = name;
13.       this.price = price;
14.    }
15.
16.    public String getName() {
17.       return name;
18.    }
19.
20.    public void setName(String name) {
21.       this.name = name;
22.    }
23.
24.    public double getPrice() {
25.       return price;
26.    }
27.
28.    public void setPrice(double price) {
29.       this.price = price;
30.    }
31. }
```

A. id
B. name
C. type
D. param
E. property
F. reqParam
G. attribute
Answer: B, D, E

Plot No : 202, IInd Floor ,HUDA Maitrivanam,Ameerpet, Hyd-5000038.
www.durgasoft.com

Q13. Given:
1. package com.example;
2.
3. public abstract class AbstractItem {
4. private String name;
...

13. }
Assume a concrete class com.example.ConcreteItem extends com.example.
AbstractItem. A servlet sets a session-scoped attribute called "item" that is an
instance of com.example.ConcreteItem and then forwards to a JSP page.
Which two are valid standard action invocations that expose a scripting variable to
the JSP page?(Choose two.)
A. <jsp:useBean id="com.example.ConcreteItem"
scope="session" />
B. <jsp:useBean id="item" type="com.example.ConcreteItem"
scope="session" />
C. <jsp:useBean id="item" class="com.example.ConcreteItem"
scope="session" />
D. <jsp:useBean id="item" type="com.example.ConcreteItem"
class="com.example.AbstractItem"
scope="session" />

Answer: B, C

Q14. Your web application views all have the same header, which includes the
<title> tag in the <head> element of the rendered HTML. You have decided to
remove this redundant HTML code from your JSPs and put it into a single JSP
called /WEB-INF/jsp/header.jsp. However, the title of each page is unique, so you
have decided to use a variable called pageTitle to parameterize this in the header
JSP, like this:

10. <title>${param.pageTitle}<title>

Which JSP code snippet should you use in your main view JSPs to insert the header
and pass the pageTitle variable?

A. <jsp:insert page='/WEB-INF/jsp/header.jsp'>
${pageTitle='Welcome Page'}
</jsp:insert>
B. <jsp:include page='/WEB-INF/jsp/header.jsp'>
${pageTitle='Welcome Page'}

</jsp:include>
C. <jsp:include file='/WEB-INF/jsp/header.jsp'>
${pageTitle='Welcome Page'}
</jsp:include>
D. <jsp:insert page='/WEB-INF/jsp/header.jsp'>
<jsp:param name='pageTitle' value='Welcome Page' />
</jsp:insert>
E. <jsp:include page='/WEB-INF/jsp/header.jsp'>
<jsp:param name='pageTitle' value='Welcome Page' />
</jsp:include>

Answer: E

Q15. Click the Exhibit button.

Given the JSP code:
1. <%
2. pageContext.setAttribute("product",
3. new com.example.Product("Pizza", 0.99));
4. %>
5. <%-- insert code here --%>
Which two, inserted at line 5, output the name of the product in the response?
(Choose two.)

```
 1.  package com.example;
 2.
 3.  public class Product {
 4.      private String name;
 5.      private double price;
 6.
 7.      public Product() {
 8.        this( "Default", 0.0 );
 9.      }
10.
11.      public Product( String name, double
price ) {
12.        this.name = name;
13.        this.price = price;
14.      }
15.
16.      public String getName() {
17.        return name;
18.      }
19.
20.      public void setName(String name) {
21.        this.name = name;
22.      }
23.
24.      public double getPrice() {
25.        return price;
26.      }
27.
28.      public void setPrice(double price) {
29.        this.price = price;
30.      }
31.  }
```

A. <%= product.getName() %>
B. <jsp:useBean id="product" class="com.example.Product" />
<%= product.getName() %>
C. <jsp:useBean id="com.example.Product" scope="page">
<%= product.getName() %>
</jsp:useBean>
D. <jsp:useBean id="product" type="com.example.Product"
scope="page" />
<%= product.getName() %>
E. <jsp:useBean id="product" type="com.example.Product">
<%= product.getName() %>
</jsp:useBean>

Q16.In a JSP-centric shopping cart application, you need to move a client's home address of the Customer object into the shipping address of the Order object. The address data is stored in a value object class called Address with properties for: street address, city, province, country, and postal code. Which two JSP code snippets can be used to accomplish this goal? (Choose two.)

A. <c:set var='order' property='shipAddress'
value='${client.homeAddress}' />

B. <c:set target='${order}' property='shipAddress'
value='${client.homeAddress}' />

C. <jsp:setProperty name='${order}' property='shipAddress'
value='${client.homeAddress}' />

D. <c:set var='order' property='shipAddress'>
<jsp:getProperty name='client' property='homeAddress' />
</c:store>

E. <c:set target='${order}' property='shipAddress'>
<jsp:getProperty name='client' property='homeAddress' />
</c:set>

F. <c:setProperty name='${order}' property='shipAddress'>
<jsp:getProperty name='client' property='homeAddress' />
</c:setProperty>

Answer: B, E

Q17. Click the Exhibit button.

Assume the product attribute does NOT yet exist in any scope.
Which two create an instance of com.example.Product and initialize the name and
price properties to the name and price request parameters? (Choose two.)

```
1.  package com.example;
2.
3.  public class Product {
4.      private String name;
5.      private double price;
6.
7.      public Product() {
8.          this( "Default", 0.0 );
9.      }
10.
11.     public Product( String name, double
price ) {
12.         this.name = name;
13.         this.price = price;
14.     }
15.
16.     public String getName() {
17.         return name;
18.     }
19.
20.     public void setName(String name) {
21.         this.name = name;
22.     }
23.
24.     public double getPrice() {
25.         return price;
26.     }
27.
28.     public void setPrice(double price) {
29.         this.price = price;
30.     }
31. }
```

A. <jsp:useBean id="product" class="com.example.Product" />
<jsp:setProperty name="product" property="*" />
B. <jsp:useBean id="product" class="com.example.Product" />
<% product.setName(request.getParameter("name")); %>
<% product.setPrice(request.getParameter("price")); %>
C. <jsp:useBean id="product" class="com.example.Product" />

```
<jsp:setProperty name="product" property="name"
value="${param.name}" />
<jsp:setProperty name="product" property="price"
value="${param.price}" />
D. <jsp:useBean id="product" class="com.example.Product">
<jsp:setProperty name="product" property="name"
value="${name}" />
<jsp:setProperty name="product" property="price"
value="${price}" />
</jsp:useBean>
```

Answer: A, C

Q18. Click the Exhibit button.
A session-scoped attribute, product, is stored by a servlet. That servlet then forwards to a JSP page. This attribute holds an instance of the com.example.Product class with a name property of "The Matrix" and price property of 39.95.

Given the JSP page code snippet:

1. <jsp:useBean id='product' class='com.example.Product'>
2. <jsp:setProperty name='product' property='price' value='49.95'/>
3. </jsp:useBean>
4. <%= product.getName() %> costs <%= product.getPrice() %>
What is the response output of this JSP page code snippet?

```
1.  package com.example;
2.
3.  public class Product {
4.      private String name;
5.      private double price;
6.
7.      public Product() {
8.          this( "Default", 0.0 );
9.      }
10.
11.     public Product( String name, double
price ) {
12.         this.name = name;
13.         this.price = price;
14.     }
15.
16.     public String getName() {
17.         return name;
18.     }
19.
20.     public void setName(String name) {
21.         this.name = name;
22.     }
23.
24.     public double getPrice() {
25.         return price;
26.     }
27.
28.     public void setPrice(double price) {
29.         this.price = price;
30.     }
31. }
```

A. Default costs 0.0
B. Default costs 49.95
C. Default costs 39.95
D. The Matrix costs 0.0
E. The Matrix costs 49.95
F. The Matrix costs 39.95

Answer: B

Plot No : 202, IInd Floor ,HUDA Maitrivanam,Ameerpet, Hyd-5000038.
www.durgasoft.com

Q19. Click the Exhibit button.
A servlet sets a session-scoped attribute product with an instance of com.example.Product and forwards to a JSP.
Which two output the name of the product in the response? (Choose two.)

```
1.  package com.example;
2.
3.  public class Product {
4.     private String name;
5.     private double price;
6.
7.     public Product() {
8.        this( "Default", 0.0 );
9.     }
10.
11.    public Product( String name, double
price ) {
12.       this.name = name;
13.       this.price = price;
14.    }
15.
16.    public String getName() {
17.       return name;
18.    }
19.
20.    public void setName(String name) {
21.       this.name = name;
22.    }
23.
24.    public double getPrice() {
25.       return price;
26.    }
27.
28.    public void setPrice(double price) {
29.       this.price = price;
30.    }
31. }
```

A. ${product.name}
B. <jsp:getProperty name="product" property="name" />
C. <jsp:useBean id="com.example.Product" />
<%= product.getName() %>
D. <jsp:getProperty name="product" class="com.example.Product"
property="name" />
E. <jsp:useBean id="product" type="com.example.Product">
<%= product.getName() %>
</jsp:useBean>

Answer: A, B

Q20. You are building your own layout mechanism by including dynamic content for the page's header and footer sections. The footer is always static, but the header generates the <title> tag that requires the page name to be specified dynamically when the header is imported. Which JSP code snippet performs the import of the header content?

A. <jsp:include page='/WEB-INF/jsp/header.jsp'>
<jsp:param name='pageName' value='Welcome Page' />
www.TestsNow.com
- 140 -
</jsp:include>
B. <jsp:import page='/WEB-INF/jsp/header.jsp'>
<jsp:param name='pageName' value='Welcome Page' />
</jsp:import>
C. <jsp:include page='/WEB-INF/jsp/header.jsp'>
<jsp:attribute name='pageName' value='Welcome Page' />
</jsp:include>
D. <jsp:import page='/WEB-INF/jsp/header.jsp'>
<jsp:attribute name='pageName' value='Welcome Page' />
</jsp:import>

Answer: A

Plot No : 202, IInd Floor ,HUDA Maitrivanam,Ameerpet, Hyd-5000038.
www.durgasoft.com