—USING CUSTOM TAGS

1. Which of the following elements are required for a valid `<taglib>` element in `web.xml`? (Select two)

   a `uri`
   b `taglib-uri`

c `tagliburi`

d `tag-uri`

e `location`

f `taglib-location`

g `tag-location`

h `tagliblocation`

Answers: b and f

Explanation

The `<taglib>` element is defined as follows:

```
<!ELEMENT taglib (taglib-uri, taglib-location)>
```

As you can see, both `taglib-uri` and `taglib-location` are required elements.

2. Which of the following `web.xml` snippets correctly defines the use of a tag library? (Select one)

a
```
<taglib>
    <uri>http://www.abc.com/sample.tld</uri>
    <location>/WEB-INF/sample.tld</location>
</taglib>
```

b
```
<tag-lib>
    <taglib-uri>http://www.abc.com/sample.tld</taglib-uri>
    <taglib-location>/WEB-INF/sample.tld</taglib-location>
</tag-lib>
```

c
```
<taglib>
    <taglib-uri>http://www.abc.com/sample.tld</taglib-uri>
   <taglib-location>/WEB-INF/sample.tld</taglib-location>
</taglib>
```

d
```
<tag-lib>
    <taglib>http://www.abc.com/sample.tld</taglib-uri>
    <taglib>/WEB-INF/sample.tld</taglib-location>
</tag-lib>
```

Answer: c

Explanation

The use of a tag library is defined using the `<taglib>` element:

```
<!ELEMENT taglib (taglib-uri, taglib-location)>
```

3. Which of the following is a valid `taglib` directive? (Select one)

a `<% taglib uri="/stats" prefix="stats" %>`

b `<%@ taglib uri="/stats" prefix="stats" %>`

c `<%! taglib uri="/stats" prefix="stats" %>`

d `<%@ taglib name="/stats" prefix="stats" %>`

e `<%@ taglib name="/stats" value="stats" %>`

Answer: b

Explanation

A directive starts with `<%@`, so answers a and c are invalid. A `taglib` directive requires `uri` and `prefix` attributes, so only answer b is correct.

4. Which of the following is a valid `taglib` directive? (Select one)

a `<%@ taglib prefix="java" uri="sunlib"%>`
b `<%@ taglib prefix="jspx" uri="sunlib"%>`
c `<%@ taglib prefix="jsp" uri="sunlib"%>`
d `<%@ taglib prefix="servlet" uri="sunlib"%>`
e `<%@ taglib prefix="sunw" uri="sunlib"%>`
f `<%@ taglib prefix="suned" uri="sunlib"%>`

Answer: f

Explanation

The JSP specification does not allow us to use the names `jspx`, `java`, `javax`, `servlet`, `sun`, and `sunw` as a value for the `prefix` attribute. Therefore, only answer f is valid.

5. Consider the following `<taglib>` element, which appears in a deployment descriptor of a web application:

```
<taglib>
 <taglib-uri>/accounting</taglib-uri>
 <taglib-location>/WEB-INF/tlds/SmartAccount.tld</taglib-location>
</taglib>
```

Which of the following correctly specifies the use of the above tag library in a JSP page? (Select one)

a `<%@ taglib uri="/accounting" prefix="acc"%>`
b `<%@ taglib uri="/acc" prefix="/accounting"%>`
c `<%@ taglib name="/accounting" prefix="acc"%>`
d `<%@ taglib library="/accounting" prefix="acc"%>`
e `<%@ taglib name="/acc" prefix="/accounting"%>`

Answer: a

Explanation

The `taglib` directive contains two attributes, `uri` and `prefix`:

- `uri`: The value of the `uri` attribute in a JSP page must be the same as the value of the `<taglib-uri>` subelement of the `<taglib>` element in `web.xml`. If the entries in `web.xml` are not used, then the value of the `uri` attribute in a JSP page can directly point to the TLD file using a root-relative URI, as in `uri="/WEB-INF/tlds/SmartAccount.tld"`.
- `prefix`: This can be any string allowed by the XML naming specification. It is similar to an alias and is used in the rest of the page to refer to this tag library.

6. You are given a tag library that has a tag named `printReport`. This tag may accept an attribute, `department`, which cannot take a dynamic value. Which of the following are correct uses of this tag? (Select two)

a `<mylib:printReport/>`

b `<mylib:printReport department="finance"/>`

c `<mylib:printReport attribute="department" value="finance"/>`

d
```
<mylib:printReport attribute="department"
     attribute-value="finance"/>
```

e
```
<mylib:printReport>
     <jsp:attribute name="department" value="finance" />
</mylib:printReport>
```

Answers: a and b

Explanation

Answer a is correct because the `department` attribute is not required. Answer b is syntactically correct, but the rest of the answers are syntactically wrong.

7. You are given a tag library that has a tag named `getMenu`, which requires an attribute, `subject`. This attribute can take a dynamic value. Which of the following are correct uses of this tag? (Select two)

a `<mylib:getMenu />`

b `<mylib:getMenu subject="finance"/>`

c
```
<% String subject="HR";%>
<mylib:getMenu subject="<%=subject%>"/>
```

d `<mylib:getMenu> <jsp:param subject="finance"/> </mylib:getMenu>`

e
```
<mylib:getMenu>
     <jsp:param name="subject" value="finance"/>
</mylib:getMenu>
```

Answers: b and c

Explanation

Answer a is wrong because `subject` is a required attribute (as the question states). Answer b is correct because a static value can be specified for an attribute that takes a dynamic value (but the reverse is not true). Answer c is correct because the `subject` attribute takes a dynamic value. Answers d and e do not make sense.

8. Which of the following is a correct way to nest one custom tag inside another? (Select one)

a
```
<greet:hello>
    <greet:world>
    </greet:hello>
</greet:world>
```

b
```
<greet:hello>
    <greet:world>
    </greet:world>
</greet:hello>
```

c `<greet:hello`
```
    <greet:world/>
/>
```
d `<greet:hello>`
```
    </greet:hello>
    <greet:world>
</greet:world>
```

Answer: b

Explanation

The inner tag should exist completely within the outer tag; therefore, answer a is not valid. We cannot use a tag as an attribute to another tag. Thus, answer c is incorrect. Answer d does not have any kind of nesting at all.

9. Which of the following elements can you use to import a tag library in a JSP document? (Select one)

a `<jsp:root>`
b `<jsp:taglib>`
c `<jsp:directive.taglib>`
d `<jsp:taglib.directive>`
e We cannot use custom tag libraries in XML format.

Answer: a

Explanation

In the XML syntax, the tag library information is included in the `<jsp:root>` element:

```
<jsp:root
   xmlns:jsp="http://java.sun.com/JSP/Page"
   xmlns:test="sampleLib.tld"
   version="1.2" >

   ...JSP PAGE...

</jsp:root>
```

10. Using `c` to represent the JSTL library, which of the following produces the same result as `<%= var %>`? (Select one)

a `<c:set value=var>`
b `<c:var out=${var}>`
c `<c:out value=${var}>`
d `<c:out var="var">`
e `<c:expr value=var>`

Answer: c

Explanation

JSTL provides the `out` tag to display a value in the JSP. Therefore, answers a, b, and e are incorrect. If the value corresponds to a variable, then the variable must be enclosed in `${…}`. Answer d is incorrect, and c is the correct choice.

11. Which attribute of `<c:if>` specifies the conditional expression? (Select one)

a `cond`
b `value`
c `check`
d `expr`
e `test`

Answer: e

Explanation

The `<c:if>` tag contains only one attribute, and this attribute represents the conditional expression that will be evaluated. This expression must be enclosed with single quotes, and it must be set equal to the `test` attribute. Answers a through d are incorrect, and e is the correct choice.

12. Which of the following JSTL `forEach` tags is valid?

a `<c:forEach varName="count" begin="1" end="10" step="1">`
b `<c:forEach var="count" begin="1" end="10" step="1">`
c `<c:forEach test="count" beg="1" end="10" step="1">`
d `<c:forEach varName="count" val="1" end="10" inc="1">`
e `<c:forEach var="count" start="1" end="10" step="1">`

Answer: b

Explanation

The JSTL `forEach` tag works like the `for` statement in Java. One major difference is the loop variable, which has to be `var`. Therefore, answers a, c, and d are incorrect. When executed, `var` starts at the `begin` value, and proceeds to the `end` value in increments of `step`. Therefore, e is incorrect and b is the correct value.

13. Which tags can be found in a JSTL `choose`? (Select two)

a `case`
b `select`
c `check`
d `when`
e `otherwise`

Answers: d and e

Explanation

The JSTL `choose` tag works like the `switch…case` structure in Java. However, each option is represented by `when`, and the default option is given by `otherwise`. Therefore, answers d and e are correct.