Persistence

persistence

persistence unit

entity class

annotations

entity manager

CRUD

Suggestions:

- Study notes
- In Netbeans, create an entity class for a database table and study the resulting code

Assume the Java web application MyApp uses persistence to access a database table. Here is part of a Java class that was generated by Netbeans.

8. (6 pts) The following code is from an entity class that was generated by Netbeans.

```
@Entity
@Table(name = "USERS")
@NamedQueries({@NamedQuery(name = "Users.findAll", query =
      "SELECT u FROM Users u"),
      @NamedQuery(name = "Users.findByUsername", query =
      "SELECT u FROM Users u WHERE u.username = :username"),
      @NamedQuery(name = "Users.findByPassword", query =
      "SELECT u FROM Users u WHERE u.password = :password"),
      @NamedQuery(name = "Users.findByEmailAddress", query =
      "SELECT u FROM Users u WHERE u.emailAddress = :emailAddress"),
      @NamedQuery(name = "Users.findByUserId", query =
      "SELECT u FROM Users u WHERE u.userId = :userId")})
```

```
public class Users implements Serializable
{
	private static final long serialVersionUID = 1L;
	@Column(name = "USERNAME")
	private String username;
	@Column(name = "PASSWORD")
	private String password;
	@Column(name = "EMAIL_ADDRESS")
	private String emailAddress;
	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	@Basic(optional = false)
	@Column(name = "USER_ID")
	private Integer userId;
```

(a) What is the name of the corresponding database table? USERS

(b) How many columns does the database table have? 4

(c) What is the primary key for the table? USER_ID

9. (8 pts) In the term CRUD,

(a) The C corresponds to the SQL statement

A. DELETE

B. INSERT

C. SELECT

D. UPDATE

(b) R corresponds to the SQL statement

A. DELETE

B. INSERT

C. SELECT

D. UPDATE

(c) U corresponds to the SQL statement

A. DELETE

B. INSERT

C. SELECT

D. UPDATE

(d) D corresponds to the SQL statement

A. DELETE

B. INSERT

C. SELECT

D. UPDATE

10. (8 pts) In a JpaController class, assume em is an instance of an EntityManager. What is the purpose of each of the following expressions?

(a) em.remove(dog);

A. save a new row in the DOG table

B. edit an existing row in the DOG table

C. delete a row in the DOG table

D. retrieve all rows of the DOG table

(b) em.createQuery("select object(o) from Dog as o").getResultList();

A. save a new row in the DOG table

B. edit an existing row in the DOG table

C. delete a row in the DOG table

D. retrieve all rows of the DOG table

(c) em.merge(dog);

A. save a new row in the DOG table

B. edit an existing row in the DOG table

C. delete a row in the DOG table

D. retrieve all rows of the DOG table

(d) em.persist(dog);

A. save a new row in the DOG table

B. edit an existing row in the DOG table

C. delete a row in the DOG table

D. retrieve all rows of the DOG table

********** Part 3 Web Services **********

11. (10 pts) Define a class **Calculator** that implements a web service. The web service has an operation named **inc** that has one int parameter **n**. The operation returns the value n + 1. ***Write the complete class definition***.

```
@WebService()
public class Calculator
{
    @WebMethod(operationName = "inc")
    public int inc(@WebParam(name = "n") int n)
    {
        return n + 1;
    }
}
```

```
// Missing annotation
@Table(name = "STUDENT")
public class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "STUDENTID")
    private Integer studentid;
    @Column(name = "FIRSTNAME")
    private String firstname;
    @Column(name = "LASTNAME")
```

```
    private String lastname;

    @Column(name = "MAJOR")

    private String major;

    @Column(name = "GPA")

    private Double gpa;
```

Questions 1-4 refer to this code.

1. The name of the corresponding database table is _____STUDENT_____.

2. The corresponding database table has how many columns? _____5_____

3. Which field is the primary key? _____STUDENTID_____

4. The annotation that is missing at the beginning of the class definition is _____@Entity_____.

5. EclipseLink, Hibernate, and TopLink are examples of

   A. User interface builders

   B. Entity classes

   C. Persistence providers

   D. Data sources

6. In the term CRUD, the C corresponds to the SQL statement

   A. DELETE

   B. INSERT

   C. SELECT

   D. UPDATE

7. In the term CRUD, the R corresponds to the SQL statement

   A. DELETE

   B. INSERT

   C. SELECT

   D. UPDATE

8. In the term CRUD, the U corresponds to the SQL statement

   A. DELETE

   B. INSERT

   C. SELECT

D. UPDATE

9. In the term CRUD, the D corresponds to the SQL statement

A. DELETE

B. INSERT

C. SELECT

D. UPDATE

10. When using persistence, we use an object that has methods find(), createQuery(), and persist(). These methods are used to perform operations on the database. The object is an instance of what class?

A. DataModel

B. EntityManager

C. Facade

D. Persistence

SOAP-Based Web Services

web service

service-oriented architecture

SOAP

JAX-WS

WSDL, including schema

annotations

defining a class to implement a web service

web service that accesses a database

web service client

web service tester

Suggestions:

- Study the notes
- Study the code for the Calculator, DogInfo, and CatInfo web services
- Study the client for the ignyte.com web service
- Create a simple web service and client and study the code and wsdl file

RESTful Web Services

resources identified by URIs

HTTP methods

RESTful web service that gives access to rows of a database table

annotations

resource class (like CatResource) that implements operations

MovieTheater example, including use of Google maps

Suggestions:

- Study the notes
- Do the following tutorial and study the code
  http://netbeans.org/kb/docs/websvc/rest.html

11. In service-oriented architecture, the client is called the

    A. service contract

    B. service consumer

    C. service provider

    D. service registry

12. JAX-WS is used in what type of web service?

    A. RESTful

B. SOAP-based

13. RESTful web services use a WSDL file.

A. true

B. false

14. In RESTful web services, resources have URIs.

A. true

B. false

15. RESTful web services use HTTP operations.

A. true

B. false

16. The code below defines a web service.

```
/* Missing annotation */
public class Compute
{
    /* Missing Annotation */(operationName = "incMult")
    public double incMult(
            @WebParam(name = "a") double a,
```

```
                @WebParam(name = "x") double x)
        {
            return a * (x + 1);
        }
    }
```

The annotation that is missing in line 1 is ____ @WebService() _____.

17. In the code shown in Problem 16, the annotation that is missing in line 4 is

_____ @WebMethod _____.

18. Here is part of the schema used in the wsdl file for the web service defined in Problem 16:

```
<xs:complexType name="incMult">
  <xs:sequence>
    <xs:element name="a" type="xs:double" />
    <xs:element name="x" type="xs:double" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="incMultResponse">
  <xs:sequence>
    <xs:element name="return" type="xs:double" />
  </xs:sequence>
</xs:complexType>
```

Suppose the following request is sent to the web service:

```
<incMult>
    <a>2.0</a>
    <x>3.0</x>
</incMult>
```

Fill in the missing portion of the resulting response below.

```
<incMultResponse>

    <return>8.0</return>

</incMultResponse>
```

19. The code below is from a client for the Compute web service (see Problem 16). Complete the last line by writing an expression that invokes the web service operation supplying arguments 3.5 and 8.2.

```
ComputeService service = new ComputeService();

Compute port = service.getComputePort();

double result = port.incMult(3.5, 8.2);
```

20. List two of the four HTTP methods:

_____GET_POST_____ _____DELETE_PUT___

21. The CatResource class implements a RESTful web service that gives access to the database table CAT. The web service method that handles requests for a Cat object must have what annotation?

A. @WebMethod

B. @Entity

C. @GET

D. @Retrieve

22. Assume the Cat database table contains the following rows:

CATID	CATNAME	CATAGE
10	Albert	14
20	Maude	10
30	Moskowitz	8
40	Garp	6
50	Tiger	3

In the Cat RESTful web service, the URL for the Cat table is

http://localhost:8080/RESTfulCatApp/resources/cats/

What is the URL of Maude's record in the table?

http://localhost:8080/RESTfulCatApp/resources/cats/20/

23. In the Movie Theater Map application, the address of the selected movie theater is obtained from _____ using a RESTful web service.

A. Google

B. a local database table

C. ignyte.com

D. fandango.com

24. In the Movie Theater Map application, the latitude and longitude of the selected movie theater is obtained from _____ using a RESTful web service.

A. Google

B. a local database table

C. pinpoint.org

D. geo.com

25. Clients for a SOAP-based web service must be implemented using the same programming language that was used to implement the web service.

A. true

B. false

HTML

Basic structure of an HTML document: <html>, <head>, <body>, <title>

Headings and paragraphs: <h1>, <h2>, <p>

Lists: <ul>, <ol>,

Tables: <table>, <thead>, <tbody>, <tr>, <th>, <td>

Links to other pages: <a>

Images: <img>

CSS

Linking a stylesheet to a web page: <link>

Colors: background-color, color

Fonts: font-style, font-weight, font-family

Borders: border, border-style, border-width, border-color

HTML and css

1. The <title> tag is contained inside which tag?

   A. <head>

   B. <body>

2. The text in an <h2> tag is displayed in a larger size font than the text in an <h4> tag.

   A. True

   B. False

3. The HTML tag for a paragraph is

   A. <par>

   B. <p>

   C. <pg>

4. Which type of list uses bullets?

   A.

   B.

5. Assume a table contains 3 rows and 7 columns and there are no headings. How many <td> elements does the table contain?

   3 * 7 = 21

6. The tag used to link to another web page is

   A.

   B. <a>

   C. <page>

D. <p>

7. The <img> tag uses which of the following attributes?

A. href

B. src

8. To assign an id to a word contained in a paragraph so we can style it, we use which tag?

A.
B. <td>
C.

9. To assign an id to a section of a web page so we can style it, we use which tag?

A. <th>

B.

C. <s>

10. To create a dropdown list in a web page, we use which of the following tags?

A.

B. <dropdown>

C. <select>

11. The items in a dropdown list are included in what tag?

A. <option>

B. <choice>

C. <item>

12. HTML5 has a tag for including audio in a web page.

A. True

B. False

13. The content of a web page is specified using

A. CSS

B. HTML

C. JavaScript

14. A Cascading Style Sheet file can only be used with a single web page.

A. True

B. False

15. Write CSS code to set the color of the text in all <h2> elements to blue.

```
h2 { color: blue; }
```

16. Write CSS code to put a green border around all paragraph elements.

```
p { border: solid green; }
```

17. Assume a web page contains an element with id = "hi". Write CSS code to display this element in the color yellow.

```
#hi { color: yellow; }
```

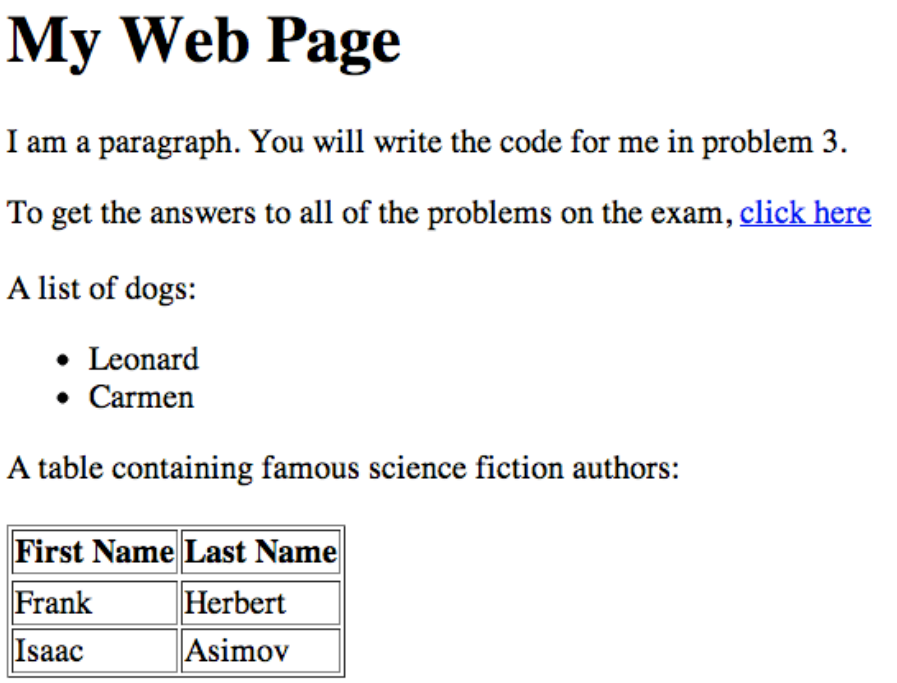
18. Responsive web design uses what type of sizes?

A. Absolute

B. Relative

19. Assume the default font size is 20px. Then a font specified to be of size 2.5em is what size in pixels?

20 * 2.5 = 50px



My Web Page

I am a paragraph. You will write the code for me in problem 3.

To get the answers to all of the problems on the exam, click here

A list of dogs:

- Leonard
- Carmen

A table containing famous science fiction authors:

First Name	Last Name
Frank	Herbert
Isaac	Asimov

1. (6 pts) Write the HTML code (including the necessary tags) to set the title of the web page to "Exam One".

```
<title>Exam One</title>
```

2. (6 pts) Write the HTML code (including the necessary tags) for the heading “My Web Page”.

```
<h1>My Web Page</h1>
```

3. (6 pts) Write the HTML code (including the necessary tags) for the paragraph.

```
<p>I am a paragraph.  You will write the code for me in problem 3.</p>
```

4. (6 pts) Write the HTML code (including the necessary tags) for the list of dogs.

```
<ul>
   <li>Leonard</li>
   <li>Carmen</li>
</ul>
```

. (10 pts) Write the HTML code (including the necessary tags) for the table containing names of science fiction authors.

```
<table border="1">
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Frank</td>
      <td>Herbert</td>
    </tr>
    <tr>
```

```
      <td>Isaac</td>
      <td>Asimov</td>
    </tr>
  </tbody>
</table>
```

6. (5 pts) Write the HTML code (including the necessary tags) to add a picture to the web page. Assume the name of the picture file is "mynewdog.jpg" and the file is located in the same directory as the .html file for the web page.

```
<img src="mynewdog.jpg" alt="My New Dog" />
```

7. (6 pts) Write the HTML code (including the necessary tags) for including the link click here to another web page. Assume the URL of the other web page is

http://cshelp.org/exam1key.html

```
<a href=http://cshelp.org/exam1key.html>click here</a>
```

8. (5 pts) Write CSS code to set the background color of the page to yellow.

```
body {
 background-color: yellow;
}
```

9. (5 pts) Write CSS code set the color of the text in the paragraph to red.

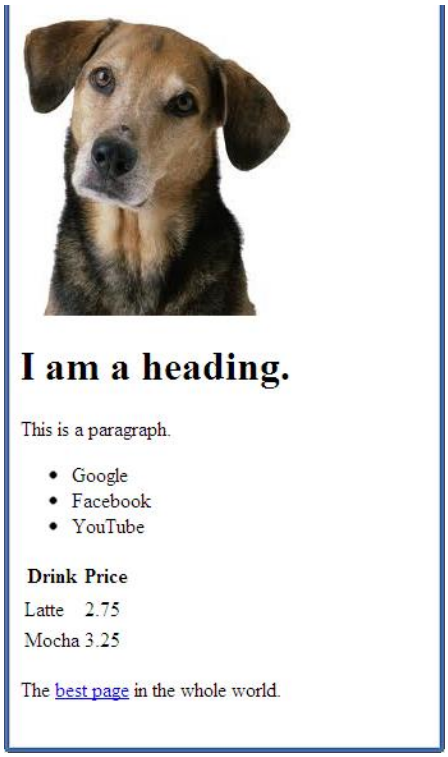
```
p {
  color: red;
}
```

10. (a) (5 pts) Assume a web page contains an element with id = "mydog". Write CSS code to display this element in the color green.

```
#mydog {
  color: green;
}
```

(b) (5 pts) Assume a web page contains one or more elements that have class="abc". Write CSS code to display all of these elements in blue.

```
.abc {
  color: blue;
}
```



1. (6 pts) Write the HTML code (including the necessary tags) for the heading "I am a heading".

```
<h1>I am a heading.</h1>
```

2. (6 pts) Write the HTML code (including the necessary tags) for the paragraph containing the text "This is a paragraph".

```
<p>
    This is a paragraph.
</p>
```

3. (6 pts) Write the HTML code (including the necessary tags) for the list containing the three items Google, Facebook, and YouTube.

```
<ul>
    <li>Google</li>
    <li>Facebook</li>
    <li>YouTube</li>
</ul>
```

4. (6 pts) Write the HTML code (including the necessary tags) for including the picture of a dog in the web page. Assume the name of the picture file is dog.jpg and the file is located in the same directory as the .html file for the web page.

```
<img src="dog.jpg" alt="A picture of a dog" />
```

5. (6 pts) Write the HTML code (including the necessary tags) for including the link best page to another web page. Assume the URL of the other page is

http://wonderfulstuff.com/greatpage.html

```
<a href="http://wonderfulstuff.com/greatpage.html">
   best page
</a>
```

6. (10 pts) Write the HTML code (including the necessary tags) for the table containing information about drinks and their prices. Be sure to include the headings for the columns.

```
<table>
    <thead>
        <tr>
            <th>Drink</th>
            <th>Price</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Latte</td>
            <td>2.75</td>
        </tr>
        <tr>
            <td>Mocha</td>
            <td>3.25</td>
        </tr>
    </tbody>
</table>
```

7. (5 pts) Write CSS code to set the background color of the page to yellow.

```
body {
    background-color: yellow;
}
```

8. (5 ps) Write CSS code to set the color of the heading to red.

```
h1 {
    color: red;
}
```

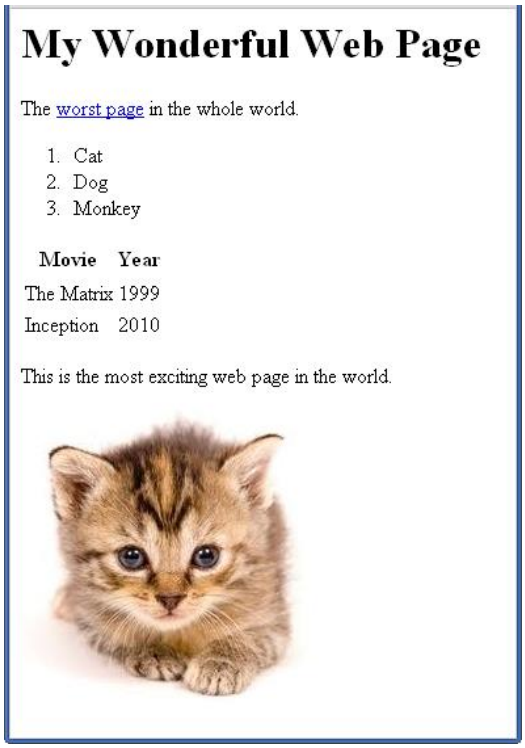
9. (5 pts) Write CSS code to put a black border around the paragraph.

```
p {
    border: 1px solid black;
}
```

10. (5 pts) Write CSS code to display all text in the table (including the column headings) in italics.

```
table {
    font-style: italic;
```

```
}
```



1. (6 pts) Write the HTML code (including the necessary tags) for including the link worst page to another web page. Assume the URL of the other page is

   http://nerd.com/badpage.html

```
<a href="http://nerd.com/badpage.html">worst page</a>
```

2. (6 pts) Write the HTML code (including the necessary tags) for the heading "My Wonderful Web Page".

```
<h1>My Wonderful Web Page</h1>
```

(6 pts) Write the HTML code (including the necessary tags) for the list containing the three items Cat, Dog, and Monkey.

```
<ol>
    <li>Cat</li>
```

```
    <li>Dog</li>
    <li>Monkey</li>
</ol>
```

4. (6 pts) Write the HTML code (including the necessary tags) for the paragraph containing the text "This is the most exciting web page in the world".

```
<p>
    This is the most exciting web page in the world.
</p>
```

5. (6 pts) Write the HTML code (including the necessary tags) for including the picture of a cat in the web page. Assume the name of the picture file is cat.jpg and the file is located in the same directory as the .html file for the web page.

```
<img src="cat.jpg" alt="A picture of a cat" />
```

6. (10 pts) Write the HTML code (including the necessary tags) for the table containing information about movies and years. Be sure to include the headings for the columns.

```
<table>
    <thead>
        <tr>
            <th>Movie</th>
            <th>Year</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>The Matrix</td>
            <td>1999</td>
```

```
        </tr>
        <tr>
            <td>Inception</td>
            <td>2010</td>
        </tr>
    </tbody>
</table>
```

7. (5 pts) Write CSS code to set the background color of the page to orange.

```
body {
    background-color: orange;
}
```

8. (5 pts) Write CSS code to display the text in the heading in italics.

```
h1 {
    font-style: italic;
}
```

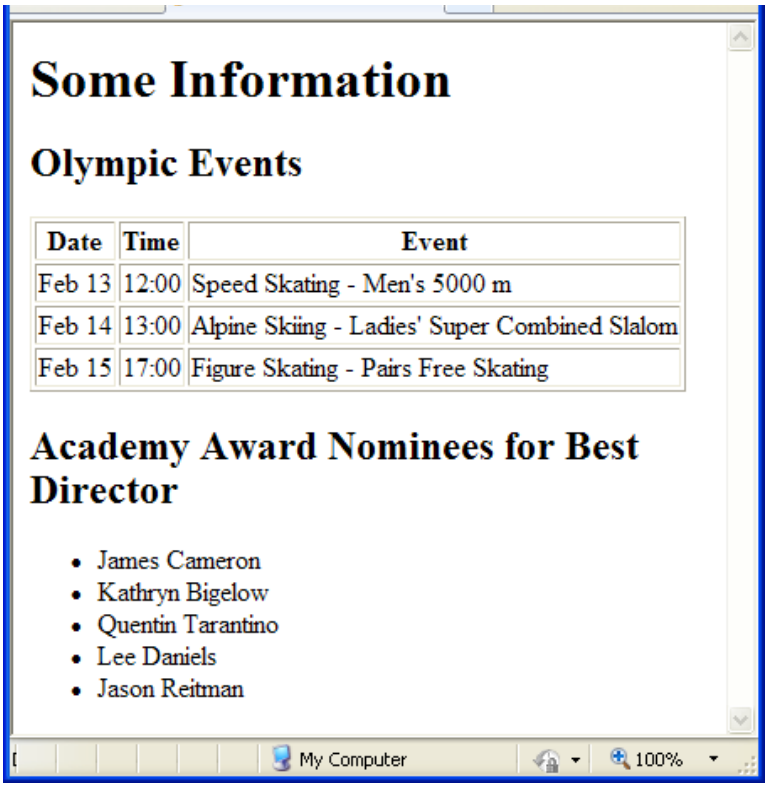
9. (5 pts) Write CSS code to put a green border around the paragraph.

```
p {
    border: 1px solid green;
}
```

10. (5 pts) Write CSS code to display all text in the table (including the column headings) in the color brown.

```
table {
    color: brown;
}
```

1. (12 pts) Complete the html code for this web page. You may continue onto the back of this page.




```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>
      Problem 1
    </title>
    <meta http-equiv="Content-Type"
          content="text/html; charset=UTF-8">
  </head>
  <body>
```

```
<h1>Some Information</h1>
    <h2>Olympic Events</h2>
    <table border="1">
        <thead>
            <tr>
                <th>Date</th>
                <th>Time</th>
                <th>Event</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Feb 13</td>
                <td>12:00</td>
                <td>Speed Skating - Men's 5000 m</td>
            </tr>
            <tr>
                <td>Feb 14</td>
                <td>13:00</td>
                <td>Alpine Skiing - Ladies' Super Combined Slalom</td>
            </tr>
            <tr>
                <td>Feb 15</td>
                <td>17:00</td>
                <td>Figure Skating - Pairs Free Skating</td>
            </tr>
        </tbody>
    </table>
```

```
    <h2>Academy Award Nominees for Best Director</h2>
    <ul>
        <li>James Cameron</li>
        <li>Kathryn Bigelow</li>
        <li>Quentin Tarantino</li>
        <li>Lee Daniels</li>
        <li>Jason Reitman</li>
    </ul>
  </body>
</html>
```

2. (8 pts) For the web page shown in Problem 1, write the code that would be included in a cascading style sheet to

(a) display the directors' names in italic letters

```
li {
   font-style: italic;
}
```

(b) display the "Some Information" heading in red letters

```
h1 {
   color: red;
}
```

(c) set the background color of the page to blue

```
body {
   background-color: blue;
}
```

(d) set the background color of the table to yellow

```
table {
   background-color: yellow;
}
```

Javascript

20. JavaScript is a subset of Java.

A.True

B. False

21. Assume a paragraph in a web page has id="hi". Write JavaScript code to set the background color of the paragraph to blue.

```
document.getElementById("hi").style.backgroundColor = "blue";
```

22. Assume we have the following input box in a web page:

<input type="text" id="username">

Write JavaScript code to assign the value the user enters in the box to the variable person.

```
var person = document.getElementById("username").value;
```

23. Assume we have the following code in a web page:

<input type="radio" name="size" id="small">Small

<input type="radio" name="size" id="large">Large

Write JavaScript code to assign the value 1.25 to the variable price if the user selected the small radio button and assign the value 2.05 to the variable price if the user selected the large radio button.

```
if (document.getElementById("small").checked) {
   price = 1.25;
```

```
} else if (document.getElementById("large").checked) {
    price = 2.05;
}
```

24. Assume we have the following tag in a web page:

<img id="myphoto" src="carmen.jpg" alt="dog photo">

Write JavaScript code to change the image to "leonard.jpg".

```
document.getElementById("myphoto").src = "leonard.jpg";
```

1. (5 pts) Write JavaScript code to set the background color of the web page to blue.

```
document.getElementById("ticketPage").style.backgroundColor = "blue";
```

2. (5 pts) Write JavaScript code to change the image to "dog.jpg".

```
document.getElementById("airlineLogo").src = "dog.jpg";
```

3. (5 pts) Write JavaScript code to assign the value the user entered in the passenger name field to the variable passenger.

```
var passenger = document.getElementById("pName").value;
```

4. (5 pts) Write JavaScript code to assign the value selected by the user in the dropdown list to the variable city.

```
var city = document.getElementById("destination").value;
```

5. (5 pts) Write JavaScript code to

- Assign the value "t" to the variable drink if the user selected the tea radio button.
- Assign the value "p" to the variable drink if the user selected the pepsi radio button.

```
if (document.getElementById("tea").checked) {
  drink = "t";
}
else if (document.getElementById("pepsi").checked) {
  drink = "p";
}
```

6. (5 pts) Write JavaScript code that adds 2.50 to the variable ticketPrice if the peanuts checkbox is checked.

```
if (document.getElementById("peanuts").checked) {
  ticketPrice += 2.50;
```

```
}
```

7. (5 pts) Write JavaScript code to add the motto "All of our pilots are over the age of 12" to the unordered list of mottos.

```
var item = document.createElement("li");
item.appendChild(document.createTextNode("All of our pilots are over the age of 12"));
document.getElementById("mottos").appendChild(item);
```

8. (5 pts) Write JavaScript code to insert the text "Be Brave" to the paragraph at the end of the web page.

```
document.getElementById("message").innerHTML = "Be Brave";
```

9. (5 pts) Write JavaScript code to assign the function reserveSeat as the event handler for clicking the button.

```
document.getElementById("purchase").onclick = reserveSeat;
```

25. Assume we have the following code in a web page:

```
<input type="button" id="checkout" value="Check Out">
```

Write JavaScript code to assign the function ProcessOrder as the event handler for clicking the button.

```
window.onload = function() {
   document.getElementById("checkout").onclick = ProcessOrder;
}
```

```
<body id="myPage">

   Age: <input type="text" id="age" /><br />

   Category: <input type="text" id="category" disabled="disabled" /><br />

   <img id="dogImage" src="young.jpg" alt="Dog Picture" /><br />

   <select name="greeting" id="greeting">
     <option value="hi">Hello</option>
     <option value="bye">Goodbye</option>
   </select>
   <br />

   <input type="radio" name="size" id="smallSize" value="small" />
     Small
   <br />
   <input type="radio" name="size" id="largeSize" value="large" />
     Large
   <br />

   <input type="checkbox" name="salt" id="salt" value="yes" />
     Salt
   <br />
   <input type="checkbox" name="pepper" id="pepper" value="yes" />
     Pepper
   <br />

   <ol id="flavors">
     <li>Vanilla</li>
     <li>Chocolate</li>
     <li>Strawberry</li>
   </ol>

   <input type="button" id="theButton" value="Click Here" />

   <p id="message"></p>

 </body>
```

1. (a) (5 pts) Write JavaScript code to assign the value the user entered in the age input box to the variable ageString.

```
var ageString = document.getElementById("age").value;
```

(b) (4 pts) Write JavaScript code to convert the value stored in ageString to an integer and store it in the variable ageInt.

```
var ageInt = parseInt(ageString);
```

(c) (8 pts) Write JavaScript code to set the text in the category input box to "Child" if the value in ageInt is less than 18, and to "Adult" otherwise.

```
if (ageInt < 18) {
  document.getElementById("category").value = "Child";
} else {
  document.getElementById("category").value = "Adult";
}
```

2. (5 pts) Write JavaScript code to assign the function doCalculations as the event handler for clicking the button.

```
document.getElementById("theButton").onclick = doCalculations;
```

3. (5 pts) Write JavaScript code to assign the value selected by the user in the dropdown list to the variable howdy.

```
var howdy = document.getElementById("greeting").value;
```

4. (8 pts) Write JavaScript code to assign the variable drinkSize the value 16 or 24, depending upon whether the user clicked on the small or large radio button.

```
var drinkSize;
if (document.getElementById("smallSize").checked) {
  drinkSize = 16;
} else if (document.getElementById("largeSize").checked) {
  drinkSize = 24;
}
```

5. (8 pts) Write JavaScript code to assign the variable seasoning the value "s" if only salt is checked, "p" if only pepper is checked, "sp" if both are checked, and "" if neither is checked.

```
var seasoning = "";
if (document.getElementById("salt").checked) {
  seasoning = seasoning + "s";
}
if (document.getElementById("pepper").checked) {
  seasoning = seasoning + "p";
}
```

6. (5 pts) Write JavaScript code to insert the text "This is the bottom of the page." in the paragraph element at the end of the web page.

```
document.getElementById("message").innerHTML = "This is the bottom of the page.";
```

7. (5 pts) Write JavaScript code to change the image to old.jpg.

```
document.getElementById("dogImage").src = "old.jpg";
```

8. (5 pts) Write JavaScript code to change the background color of the web page to blue.

```
document.getElementById("myPage").style.backgroundColor = "blue";
```

9. (8 pts) Write JavaScript code to add the flavor Caramel to end of the ordered list.

```
var item = document.createElement("li");
item.appendChild(document.createTextNode("Caramel"));
document.getElementById("flavors").appendChild(item);
```

Jquery

1. jQuery is a collection of Java methods.

   A. True

   B. False

2. Assume a web page contains this HTML:

```
<p id="hello"></p>
```

   Write jQuery code to set the content of the paragraph to the string "Greetings".

```
$("#hello").html("Greetings");
```

3. Assume a web page contains this HTML:

```
<input type="text" id="user">
```

   Write jQuery code to assign the value the user entered in this input box to the variable username.

var username = $("#user").val();

4. Assume a web page contains this HTML:

This is a paragraph.

Write jQuery code to hide this element.

$("#hi").hide();

1. jQuery is a JavaScript library.

A. True

B. False

2. Assume a web page contains this HTML:

<button id="hello">Greetings</button>

Write jQuery code to change the label on the button to "Goodbye".

$("#hello").html("Goodbye");

3. Assume a web page contains this HTML:

<input type="text" id="user">

Write jQuery code to assign the value the user entered in this input box to the variable username.

var username = $("#user").val();

4. Assume a web page contains this HTML:

This is a paragraph.

Write jQuery code to remove this element from the web page.

$("#hi").remove();

1. jQuery is used for server-side programming.

A. True

B. False

```
<body id="ticketPage">
    <img id="airlineLogo" src="bearcat.jpg" alt="Airline Logo"><br>
    Passenger Name: <input type="text" id="pName"><br>
    Destination<br>
    <select id="destination">
        <option value="dfw">Dallas</option>
        <option value="lax">Los Angeles</option>
        <option value="jfk">New York</option>
    </select><br>
    Drink<br>
    <input type="radio" id="tea">Tea<br>
    <input type="radio" id="pepsi">Pepsi<br>
    Snack<br>
    <input type="checkbox" id="pretzels">Pretzels<br>
    <input type="checkbox" id="peanuts">Peanuts<br>
    Airline Mottos
    <ul id="mottos">
        <li>We'll get there eventually</li>
        <li>BYOP (Bring Your Own Parachute)</li>
```

```
    </ul>
    <input type="button" id="purchase" value="Purchase Ticket">
    <p id="message"></p>
</body>
```

10. (5 pts) Write jQuery code to hide the image.

$('#airlineLogo").hide();

11. (5 pts) Write jQuery code to remove the paragraph.

$('#message').remove();

12. (5 pts) JavaScript is executed

    A. by a browser on the client side.

    B. by Glassfish on the server side.

13. (5 pts) jQuery is

    A. a JavaScript library

    B. a collection of Java methods

```
<body id="myPage">
   GPA: <input type="text" id="gpa" /><br />
   Scholarship: <input type="text" id="scholarship" disabled="disabled" />
   <br />
   <img id="dogImage" src="young.jpg" alt="Dog Picture" /><br />
   <select name="greeting" id="greeting">
     <option value="hi">Hello</option>
     <option value="bye">Goodbye</option>
   </select>
   <br />
```

```
  <input type="radio" name="size" id="smallSize" value="small" />
    Small
  <br />
  <input type="radio" name="size" id="largeSize" value="large" />
    Large
  <br />

  <input type="checkbox" name="salt" id="salt" value="yes" />
    Salt
  <br />
  <input type="checkbox" name="pepper" id="pepper" value="yes" />
    Pepper
  <br />

  <ol id="flavors">
    <li>Vanilla</li>
    <li>Chocolate</li>
    <li>Strawberry</li>
  </ol>

  <input type="button" id="theButton" value="Click Here" />

  <p id="message"></p>

</body>
```

1. (a) (5 pts) Write JavaScript code to assign the value the user entered in the gpa input box to the variable gpaString.

```
var gpaString = document.getElementById("gpa").value;
```

(b) (4 pts) Write JavaScript code to convert the value stored in gpaString to a float and store it in the variable gpaFloat.

```
var gpaFloat = parseFloat(gpaString);
```

(c) (8 pts) Write JavaScript code to set the text in the scholarship input box to "Yes" if the value in gpaFloat is greater than 3.3, and to "No" otherwise.

```
if (gpaFloat > 3.3) {
  document.getElementById("scholarship").value = "Yes";
} else {
  document.getElementById("scholarship").value = "No";
}
```

2. (5 pts) Write JavaScript code to assign the function doCalculations as the event handler for clicking the button.

```
document.getElementById("theButton").onclick = doCalculations;
```

3. (5 pts) Write JavaScript code to assign the value selected by the user in the dropdown list to the variable howdy.

```
var howdy = document.getElementById("greeting").value;
```

4. (8 pts) Write JavaScript code to assign the variable drinkSize the value 16 or 24, depending upon whether the user clicked on the small or large radio button.

```
var drinkSize;
if (document.getElementById("smallSize").checked) {
  drinkSize = 16;
} else if (document.getElementById("largeSize").checked) {
  drinkSize = 24;
}
```

5. (8 pts) Write JavaScript code to assign the variable seasoning the value "s" if only salt is checked, "p" if only pepper is checked, "sp" if both are checked, and "" if neither is checked.

```
var seasoning = "";
if (document.getElementById("salt").checked) {
  seasoning = seasoning + "s";
}
if (document.getElementById("pepper").checked) {
  seasoning = seasoning + "p";
}
```

6. (5 pts) Write JavaScript code to insert the text "I am a paragraph." in the paragraph element at the end of the web page.

```
document.getElementById("message").innerHTML = "I am a paragraph.";
```

7. (5 pts) Write JavaScript code to change the image to cat.jpg.

```
document.getElementById("dogImage").src = "cat.jpg";
```

8. (5 pts) Write JavaScript code to change the background color of the web page to red.

```
document.getElementById("myPage").style.backgroundColor = "red";
```

9. (8 pts) Write JavaScript code to add the flavor Cherry to end of the ordered list.

```
var item = document.createElement("li");
item.appendChild(document.createTextNode("Cherry"));
document.getElementById("flavors").appendChild(item);
```

JavaScript

- Use document.getElementById to obtain a reference to a web page element
- Obtain value from input element
- parseInt () and parseFloat()
- Assign value to input element
- Assign action handler function to event
- Obtain values from dropdown lists, radio buttons, and check boxes
- innerHTML property
- Arrays
- Change image in img tag
- Change background color of an element
- appendChild(), removeChild(), createTextNode(), createElement(), childNodes

jQuery

- $()

html()

show()

hide()

remove()

hover()

click()

toggleClass()

fadeOut()

XML

5. XML is case-sensitive.

   A. True

   B. False

6. XML uses namespaces to avoid

   A. Extra white space in documents

   B. Spelling errors in tag names

   C. Naming collisions in documents

7. Each XML namespace has a unique prefix that must always be used in all documents.

   A. True

   B. False

8. Assume we have defined an XML language using the DTD below.

   <?xml version="1.0" encoding="UTF-8"?>

   <!ELEMENT a (b, c+, d*)>

<!ELEMENT b (#PCDATA)>

<!ELEMENT c (#PCDATA)>

<!ELEMENT d (#PCDATA)>

The root element is

A. a

B. b

C. c

D. d

9. Assume we have defined an XML language using the DTD below.

<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT a (b, c+, d*)>

<!ELEMENT b (#PCDATA)>

<!ELEMENT c (#PCDATA)>

<!ELEMENT d (#PCDATA)>

The element b may be omitted.

A. True

B. False

10. Assume an XML schema contains this code:

<xs:sequence>

<xs:element name="s" type="xs:string" maxOccurs="unbounded"/>

</xs:sequence>

The element s can be omitted.

A. True

B. False

11. Assume an XML schema contains this code:

```
<xs:simpleType name="T">
  <xs:restriction base="xs:string">
    <xs:pattern value="[2-6]{3}(\d{2})?"/>
  </xs:restriction>
</xs:simpleType>
```

Circle all of the values below that are legal values for an element of type T.

A. 463

B. 23456

C. 45678

D. 87654

E. 765

5. XML is a Microsoft product and can only be used in a Windows environment.

A. True

B. False

6. XML uses namespaces to avoid

A. Extra white space in documents

B. Spelling errors in tag names

C. Naming collisions in documents

7. Assume we have defined an XML language using the DTD below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT a (b, c+, d*)>
<!ELEMENT b (#PCDATA)>
<!ELEMENT c (#PCDATA)>
<!ELEMENT d (#PCDATA)>
```

The element c can occur twice.

A. True

B. False

8. Assume an XML schema contains this code:

```
<xs:sequence>
  <xs:element name="s" type="xs:string" maxOccurs="unbounded"/>
</xs:sequence>
```

The element s can occur three times.

A. True

B. False

9. Assume an XML schema contains this code:

```
<xs:sequence>
  <xs:element name="s" type="xs:string" minOccurs="0"/>
</xs:sequence>
```

The element s can occur two times.

A. True

B. False

10. Assume an XML schema contains this code:

```
<xs:simpleType name="count">
  <xs:restriction base="xs:positiveInteger">
    <xs:maxInclusive value="6"/>
  </xs:restriction>
</xs:simpleType>
```

The minimum legal value of an element of type count is

A. 0

B. 1

C. 5

1. Assume an XML schema contains this code:

```
<xs:simpleType name="T">
  <xs:restriction base="xs:string">
    <xs:pattern value="[2-5]{3}(\d{2})?"/>
  </xs:restriction>
</xs:simpleType>
```

Circle all of the values below that are legal values for an element of type T.

A. 463

B. 23456

C. 45678

D. 87654

E. 765

5. XML is part of Java and can only be parsed using Java.

A. True

B. False

6. XML uses namespaces to avoid

A. Extra white space in documents

B. Spelling errors in tag names

C. Naming collisions in documents

7. Each XML namespace has a unique URI that must always be used in all documents.

A. True

B. False

8. Assume we have defined an XML language using the DTD below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT a (b, c+, d*)>
<!ELEMENT b (#PCDATA)>
<!ELEMENT c (#PCDATA)>
<!ELEMENT d (#PCDATA)>
```

The element d may be omitted.

A. True

B. False

9. Assume an XML schema contains this code:

```
<xs:sequence>
  <xs:element name="s" type="xs:string" minOccurs="0"/>
</xs:sequence>
```

The element s can occur two times.

A. True

B. False

10. Assume an XML schema contains this code:

```
<xs:simpleType name="count">
  <xs:restriction base="xs:positiveInteger">
    <xs:maxExclusive value="6"/>
  </xs:restriction>
</xs:simpleType>
```

The maximum legal value of an element of type count is

A. 0

B. 1

C. 5

D. 6

11. Assume an XML schema contains this code:

```
<xs:simpleType name="T">
  <xs:restriction base="xs:string">
    <xs:pattern value="[2-6]{2}(\d{3})?"/>
  </xs:restriction>
</xs:simpleType>
```

Circle all of the values below that are legal values for an element of type T.

A. 463

B. 23456

C. 45678

D. 87654

E. 765

1.

3. (10 pts) The student language is defined by this DTD file, named **student.dtd**.

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
    Document   : student.dtd
    Description: Describes the student language
-->

<!ELEMENT student (name, courseList)>
   <!ATTLIST student studentNumber ID #REQUIRED>
<!ELEMENT name (first, last)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT last (#PCDATA)>
<!ELEMENT courseList (course+)>
<!ELEMENT course (courseNumber, grade)>
<!ELEMENT courseNumber (#PCDATA)>
<!ELEMENT grade (#PCDATA)>
```

Write an XML file **student.xml** that uses the student language to represent the data described below.

> Leonard Poodle is a student with student number S123654. He is taking two courses: 99-888 with a grade of 93 and 11-222 with a grade of 86.

```
<student studentNumber="S123456">
  <name>
```

```
    <first>Leonard</first>
    <last>Poodle</last>
  </name>
  <courseList>
    <course>
      <courseNumber>99-888</courseNumber>
      <grade>93</grade>
    </course>
    <course>
      <courseNumber>11-222</courseNumber>
      <grade>86</grade>
    </course>
  </courseList>
</student>
```

4. (a) (2 pts) Each namespace has a unique URI.

TRUE FALSE

(b) (2 pts) Each namespace has a unique prefix that must be used in all documents that use the namespace.

TRUE FALSE

(c) (2 pts) In the namespace declaration below

```
xmlns:mml="http://www.w3.org/1998/Math/MathML"
```

the namespace prefix is _____mml_____

and the namespace URI is _http://www.w3.org/1998/Math/MathML_

5. (10 pts) Assume we want to define a language to represent information about cities. Here is a sample XML file that uses the language.

```
<!DOCTYPE cityList SYSTEM 'cities.dtd'>
<cityList>
    <city name="Dallas">
        <population>5000000</population>
        <temperature>
            <avrgHigh>82</avrgHigh>
            <avrgLow>68</avrgLow>
        </temperature>
    </city>
    <city name="Detroit">
        <population>9000000</population>
        <temperature>
            <avrgHigh>68</avrgHigh>
            <avrgLow>42</avrgLow>
        </temperature>
    </city>
</cityList>
```

Write the DTD file cities.dtd that defines this language. The XML file above must validate against the DTD you write. The city name is required and must be unique.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT cityList (city+)>
<!ELEMENT city (population, temperature)>
```

```
  <!ATTLIST city name ID #REQUIRED>

<!ELEMENT population (#PCDATA)>

<!ELEMENT temperature (avrgHigh, avrgLow)>

<!ELEMENT avrgHigh (#PCDATA)>

<!ELEMENT avrgLow (#PCDATA)>
```

6. (10 pts) Assume an XML file has the following format:

```
<cityList>
    <city>
        <name>Dallas</name>
        <population>5000000</population>
        <temperature>
            <avrgHigh>82</avrgHigh>
            <avrgLow>68</avrgLow>
        </temperature>
    </city>

    ***** more city elements *****

</cityList>
```

We want to use XSL to transform this data into a table in HTML. The list will contain the name and avrgHigh value for each city with a population greater than 250000. Write XSL code that produces this list from the data file. **Write ONLY the code that goes between the <table> and </table> tags. Everything else can be omitted.**

```
<table>
  <xsl:for-each select="cityList/city">
    <xsl:if test="population &gt; 250000">
      <tr>
        <td>
          <xsl:value-of select="name"/>
        </td>
```

```
        <td>
          <xsl:value-of select="temperature/avrgHigh"/>
        </td>
      </tr>
    </xsl:if>
  </xsl:for-each>
</table>
```

7. (10 pts) Answer the questions below concerning this definition from a schema:

```
<xsd:complexType name="Items">
  <xsd:sequence>
    <xsd:element name="item" minOccurs="1" maxOccurs="10">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="productName" type="xsd:string"/>
          <xsd:element name="quantity">
            <xsd:simpleType>
              <xsd:restriction base="xsd:positiveInteger">
                <xsd:maxExclusive value="12"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="USPrice"  type="xsd:decimal"/>
          <xsd:element ref="comment"   minOccurs="0"/>
          <xsd:element name="shipDate" type="xsd:date" minOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="partNum" type="SKU" use="required"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="SKU">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d{2}[A-Z]{2}\d{4}"/>
```

```
  </xsd:restriction>
</xsd:simpleType>
```

(a) The **item** element can be omitted.

TRUE FALSE

(b) The **item** element can occur 20 times.

TRUE FALSE

(c) The **shipDate** element is required.

TRUE FALSE

(d) The value of **quantity** can be 0.

TRUE FALSE

(e) The maximum legal value of **quantity** is ____11_____.

(f) An **item** has an attribute named __partNum_____.

(g) Give an example of a legal value for **partNum**. 38HP7249

8. Assume we are using Java to parse the xml document below.

```
<dogList>
    <dog>
        <dogName>Tex</dogName>
        <breed>Standard Poodle</breed>
        <age>6</age>
        <owner>
            <ownerName>
                <first>Mary</first>
                <last>Smith</last>
            </ownerName>
            <address>
                <city>Dallas</city>
                <state>Texas</state>
            </address>
        </owner>
    </dog>

    REMAINING dog ELEMENTS NOT SHOWN TO SAVE SPACE

</dogList>
```

(a) (3 pts) In order to extract the breed of the third dog, we would use the expression (*write the appropriate value for the first argument **below** the method call*):

```
path.evaluate("  ", doc)
```

```
"/dogList/dog[3]/breed"
```

(b) (3 pts) In order to extract the first name of the owner of the fifth dog, we would use the expression:

```
path.evaluate("  ", doc)
```

```
"/dogList/dog[5]/owner/ownerName/first"
```

(c) (3 pts) In order to extract the city of the owner of the first dog, we would use the expression:

```
path.evaluate("  ", doc)
```

```
"/dogList/dog[1]/owner/address/city"
```

XML

Basic structure of an XML document: elements, root element, subelements, attributes

Representing simple expressions in MathML: HelloMML.xml, QuadraticMML.xml

Representing simple shapes in SVG: Picture.xml

Namespaces

Naming collisions

URI for a namespace

Prefix for a namespace

xmlns attribute

DTDs

!ELEMENT, !ATTLIST, EMPTY, #PCDATA, ?, *, +, CDATA, ID, #IMPLIED, #REQUIRED

Given a DTD and some data, represent the data in XML so that it conforms to the DTD

Given an example XML document, write a DTD that specifies its format

XSL

Path expressions in XPATH

XSLT: for-each, if, value-of, sort

Schemas

Be able to read and analyze a schema

Be able to answer questions regarding the structure of XML documents that conform to the schema

(c) (2 pts) In the namespace declaration below

```
xmlns:ufo="http://www.alienVisitors.org/area51"
```

the namespace prefix is ____ufo_____

and the namespace URI is http://www.alienVisitors.org/area51

12. (10 pts) The patients language is defined by this DTD file, named patients.dtd.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT patientList (patient+)>
<!ELEMENT patient (age, weight, bloodPressure)>
   <!ATTLIST patient patientId ID #REQUIRED>
<!ELEMENT age (#PCDATA)>
<!ELEMENT weight (#PCDATA)>
<!ELEMENT bloodPressure (systolic, diastolic)>
<!ELEMENT systolic (#PCDATA)>
<!ELEMENT diastolic (#PCDATA)>
```

Write an XML file patients.xml that uses the patients language to represent the data described below. Your file must validate against the DTD file.

Eve is a patient with id P1234, age 23, weight 128, systolic blood pressure 119, and diastolic blood pressure 76.

```
<patientList>
   <patient patientId="P1234">
      <age>23</age>
      <weight>128</weight>
      <bloodPressure>
         <systolic>119</systolic>
         <diastolic>76</diastolic>
      </bloodPressure>
   </patient>
```

```
</patientList>
```

13.

```
<xsd:complexType name="Items">
  <xsd:sequence>
    <xsd:element name="item" minOccurs="1" maxOccurs="4">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="productName" type="xsd:string"/>
          <xsd:element name="quantity">
            <xsd:simpleType>
              <xsd:restriction base="xsd:positiveInteger">
                <xsd:maxExclusive value="10"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="USPrice"  type="xsd:decimal"/>
          <xsd:element ref="comment"   minOccurs="0"/>
          <xsd:element name="shipDate" type="xsd:date" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="partNum" type="SKU" use="required"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="SKU">
  <xsd:restriction base="xsd:string">
```

```
    <xsd:pattern value="\d{2}[A-Z]{3}\d{4}"/>
  </xsd:restriction>
</xsd:simpleType>
```

(a) The **item** element can be omitted.

TRUE FALSE

(b) The **item** element can occur 5 times.

TRUE FALSE

(c) The **shipDate** element is required.

TRUE FALSE

(d) The value of **quantity** can be 3.

TRUE FALSE

(e) The maximum legal value of **quantity** is ___9_____.

(f) Give an example of a legal value for **partNum**. 74CAT4912

14. (12 pts) Assume an XML file has the following format:

```
<dogList>
    <dog>
        <dogName>Tex</dogName>
        <breed>Standard Poodle</breed>
        <age>6</age>
        <owner>
            <ownerName>
                <first>Mary</first>
                <last>Smith</last>
            </ownerName>
            <address>
                <city>Dallas</city>
                <state>Texas</state>
            </address>
        </owner>
    </dog>

    REMAINING dog ELEMENTS NOT SHOWN TO SAVE SPACE

</dogList>
```

We want to use XSL to transform this data into a table in HTML. The table will contain the breed and owner's last name for each dog with an age greater than 2. Write XSL code that produces this table from the data file. **Write ONLY the code that goes between the <table> and </table> tags. Everything else can be omitted.**

NOTE: THERE IS MORE SPACE AVAILABLE ON THE NEXT PAGE

```
<table>
  <xsl:for-each select="dogList/dog">
    <xsl:if test="age &gt; 2">
      <tr>
        <td>
          <xsl:value-of select="breed"/>
        </td>
```

```
        <td>
          <xsl:value-of select="owner/ownerName/last"/>
        </td>
      </tr>
    </xsl:if>
  </xsl:for-each>
</table>
```

(c) (2 pts) In the namespace declaration below

xmlns:ape="http://www.animals.org/hominidae"

the namespace URI is **http://www.animals.org/hominidae**

and the namespace prefix is __ape_____

12. (10 pts) The faculty language is defined by this DTD file, named faculty.dtd.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT faculty (name, adviseeList)>
   <!ATTLIST faculty facultyNumber ID #REQUIRED>
<!ELEMENT name (first, last)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT last (#PCDATA)>
<!ELEMENT adviseeList (advisee+)>
<!ELEMENT advisee (studentNumber, classification)>
<!ELEMENT studentNumber (#PCDATA)>
<!ELEMENT classification (#PCDATA)>
```

Write an XML file faculty.xml that uses the faculty language to represent the data described below. Your file must validate against the DTD file.

> Donald Knuth is a faculty member with faculty number F849. He has two advisees: Student S456 who is a junior, and student S789 who is a senior.

```
<faculty facultyNumber="F849">
   <name>
      <first>Donald</first>
      <last>Knuth</last>
   </name>
   <adviseeList>
      <advisee>
       <studentNumber>S456</studentNumber>
```

```
        <classification>junior</classification>
      </advisee>
      <advisee>
        <studentNumber>S789</studentNumber>
        <classification>senior</classification>
      </advisee>
    </adviseeList>
</faculty>
```

13. (12 pts) Answer the questions below concerning this definition from a schema:

```
<xsd:complexType name="Items">
  <xsd:sequence>
    <xsd:element name="item" minOccurs="0" maxOccurs="7">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="productName" type="xsd:string"/>
          <xsd:element name="quantity">
            <xsd:simpleType>
              <xsd:restriction base="xsd:positiveInteger">
                <xsd:maxExclusive value="10"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="USPrice"  type="xsd:decimal"/>
          <xsd:element ref="comment"   minOccurs="0"/>
          <xsd:element name="shipDate" type="xsd:date" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="partNum" type="SKU" use="required"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="SKU">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[A-Z]{3}\d{2}"/>
```

```
  </xsd:restriction>
</xsd:simpleType>
```

(a) The **item** element can be omitted.

TRUE FALSE

(b) The **item** element can occur 5 times.

TRUE FALSE

(c) The **shipDate** element is required.

TRUE FALSE

(d) The value of **quantity** can be 10.

TRUE FALSE

(e) The minimum legal value of **quantity** is ____1_____.

(f) Give an example of a legal value for **partNum**. CAT27

14. (12 pts) Assume an XML file has the following format:

```
<dogList>
    <dog>
        <dogName>Tex</dogName>
        <breed>Standard Poodle</breed>
        <age>6</age>
        <owner>
            <ownerName>
                <first>Mary</first>
                <last>Smith</last>
            </ownerName>
            <address>
                <city>Dallas</city>
                <state>Texas</state>
            </address>
        </owner>
    </dog>

    REMAINING dog ELEMENTS NOT SHOWN TO SAVE SPACE

</dogList>
```

We want to use XSL to transform this data into a table in HTML. The table will contain the dog name and owner's city for each dog with an age less than 8. Write XSL code that produces this table from the data file. **Write ONLY the code that goes between the <table> and </table> tags. Everything else can be omitted.**

NOTE: THERE IS MORE SPACE AVAILABLE ON THE NEXT PAGE

```
<xsl:for-each select="dogList/dog">
    <xsl:if test="age &lt; 8">
      <tr>
        <td>
          <xsl:value-of select="dogName"/>
        </td>
        <td>
```

```
            <xsl:value-of select="owner/address/city"/>
          </td>
        </tr>
      </xsl:if>
    </xsl:for-each>
```

1. (12 pts) Answer the questions below concerning this definition from a schema:

```
<xsd:complexType name="Items">
  <xsd:sequence>
    <xsd:element name="item" minOccurs="1" maxOccurs="6">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="productName" type="xsd:string"/>
          <xsd:element name="quantity">
            <xsd:simpleType>
              <xsd:restriction base="xsd:positiveInteger">
                <xsd:maxExclusive value="5"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="USPrice"  type="xsd:decimal"/>
          <xsd:element ref="comment"   minOccurs="0"/>
          <xsd:element name="shipDate" type="xsd:date" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="partNum" type="SKU" use="required"/>
      </xsd:complexType>
    </xsd:element>
```

```
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="SKU">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d{2}[A-Z]{3}\d{2}"/>
  </xsd:restriction>
</xsd:simpleType>
```

(a) The **item** element can be omitted.

TRUE FALSE

(b) The **item** element can occur 7 times.

TRUE FALSE

(c) The **shipDate** element is required.

TRUE FALSE

(d) The value of **quantity** can be 0.

TRUE FALSE

(e) The maximum legal value of **quantity** is ___4_____.

(f) Give an example of a legal value for **partNum**. 38DFW72

12. Assume we are using Java to parse the xml document below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE student SYSTEM 'student.dtd'>
<student studentNumber="S24">
   <name>
      <first>Midge</first>
      <last>McDonald</last>
   </name>
   <courseList>
      <course>
         <courseNumber>44-100</courseNumber>
         <grade>84</grade>
      </course>
      <course>
         <courseNumber>44-200</courseNumber>
         <grade>93</grade>
      </course>
      <course>
         <courseNumber>44-123</courseNumber>
         <grade>89</grade>
      </course>
   </courseList>
</student>
```

(a) (3 pts) In order to extract the last name for the student, we would use the expression (write the appropriate value for the first argument ***below*** the method call):

```
path.evaluate("  ", doc)
```

/student/name/last

(b) (5 pts) In order to extract the course number for the first course, we would use the expression:

```
path.evaluate("  ", doc)
```

/student/courseList/course[1]/courseNumber

(c) (5 pts) In order to extract the grade for the third course, we would use the expression:

```
path.evaluate("  ", doc)
```

/student/courseList/course[3]/grade

13. (10 pts) An XML language for representing appointments is defined by the following DTD:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT appointmentList (appointment+)>
<!ELEMENT appointment (date, time, description)>
<!ELEMENT date (day, month, year)>
<!ELEMENT day (#PCDATA)>
<!ELEMENT month (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT time (start, end)>
<!ELEMENT start (#PCDATA)>
<!ELEMENT end (#PCDATA)>
<!ELEMENT description (#PCDATA)>
```

Write an XML file that uses the appointments language to represent the data described below.

There will be an ACM meeting on March 25, 2013, from 17:30 to 18:30.

```
<appointmentList>
  <appointment>
    <date>
      <day>25</day>
      <month>March</month>
      <year>2013</year>
    </date>
    <time>
      <start>17:30</start>
```

```
      <end>18:30</end>
    </time>
    <description>ACM Meeting</description>
  </appointment>
</appointmentList>
```

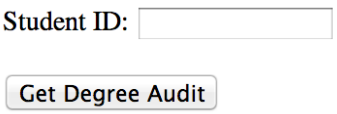
Servlets

12. Servlet code is executed

    A. By a browser on the client side

    B. By Glassfish on the server side

13. The web page shown below contains a form. When the button is clicked, the servlet degAudit handles the request.



The code for the form is shown below. Fill in the blanks to complete the code.

```
<form ___action_____="degAudit>
  Student ID:
  <input type="text" ____name_____="studentId">
  <br><br>
  <input type="_____submit_____" value="Get Degree Audit">
</form>
```

14. Write servlet code that assigns the value of the request parameter named studentID to the string variable sid.

String sid = request.getParameter("studentId");

15. Sessions are necessary because

A. HTTP is a stateless protocol.

B. The GET operation does not allow the use of request parameters.

16. The application server sends a session ID to the client using a

A. request parameter

B. submit parameter

C. cookie

16. Assume a web application has created an HttpSession object named session. Complete the code below to save a username "Leonard" so that it can be accessed thoughout the session.

session.setAttribute_____("username", "Leonard");

JavaScript

Use document.getElementById to obtain a reference to a web page element

Obtain value from input element

parseInt () and parseFloat()

Assign value to input element

Assign action handler function to event

Obtain values from dropdown lists, radio buttons, and check boxes

innerHTML property

Arrays

Change image in img tag

Change background color of an element

firstChild, nextSibling, appendChild(), removeChild(), createTextNode()

Servlets

form tag and action attribute

submit buttons

Return html for a web page

Get value of a request parameter

request.getSession()

Session attributes

AJAX

Set up a request object and send request to servlet

Use callback function to receive message from servlet

Extract data from request.responseText or request.responseXML

1. (8 pts) Complete the code in the processRequest() method of a servlet that returns a web page that displays the message Hello, Web Surfer. The title of the page must be Problem One.

```
PrintWriter out = response.getWriter();
try {
   out.println("<html>");
   out.println("<head>");
   out.println("<title>Problem One</title>");
   out.println("</head>");
   out.println("<body>");
   out.println("<h1>Hello, Web Surfer</h1>");
```

```
        out.println("</body>");
        out.println("</html>");
    } finally {
        out.close();
    }
```

In problems 2 and 3, you will write XML code to be included in the deployment descriptor for a web application. Here are some of the tags you might need to use:

<context-param>	**<description>**
<init-param>	**<listener>**
<listener-class>	**<param-name>**
<param-value>	**<servlet>**
<servlet-class>	**<servlet-name>**
<servlet-mapping>	**<url-pattern>**

2. (5 pts) Write the XML code that would be included in the deployment descriptor to define a context init parameter named **dogName** with value **Eve**.

```
<context-param>
    <param-name>dogName</param-name>
    <param-value>Eve</param-value>
</context-param>
```

3. (10 pts) Write the XML code that would be included in the deployment descriptor for a servlet with name **`Cat Servlet`**, class **`Kitty`**, and url **`/Cat`**. The servlet has an init parameter named **`catName`** with value **`Moskowitz`**.

```
<servlet>
    <servlet-name>Cat Servlet</servlet-name>
    <servlet-class>Kitty</servlet-class>
    <init-param>
        <param-name>catName</param-name>
        <param-value>Moskowitz</param-value>
    </init-param>
</servlet>
<servlet-mapping>
    <servlet-name>Cat Servlet</servlet-name>
    <url-pattern>/Cat</url-pattern>
</servlet-mapping>
```

4. (a) (4 pts) Write servlet code that would assign the value of the request parameter named **sn** to the String variable **snakeName**.

```
String snakeName = request.getParameter("sn");
```

(b) (4 pts) Write servlet code that would assign the value of the servlet init parameter named **fn** to the String variable **fishName**.

```
String fishName
= getServletConfig().getInitParameter("fn");
```

(c) (4 pts) Write servlet code that would assign the value of the context init parameter named **bn** to the String variable **birdName**.

```
String birdName
= getServletContext().getInitParameter("bn");
```

5. (5 pts) Assume a client invokes servlet S1 that sets up a session attribute as follows:

```
HttpSession session = request.getSession();
session.setAttribute("city", "New York");
```

Later the same client invokes another servlet S2. Write the code that must be included in servlet S2 to assign the value of the session attribute to the variable **cityName**.

```
HttpSession session = request.getSession();
String cityName
= (String)session.getAttribute("city");
```

6. (9 pts) Assume class Dog has the constructor shown below:

```
public Dog(String dogName, int dogAge)
```

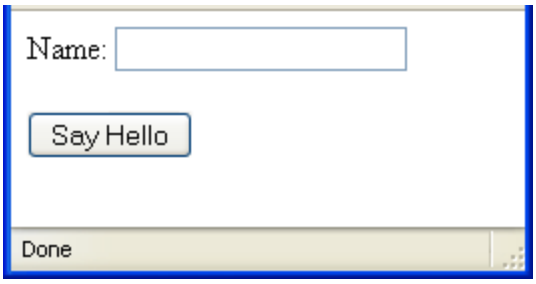
In an MVC application, the controller

- creates a Dog object with name "Carmen" and age 7,
- makes the object the value of a request attribute named **dog**
- forwards to the view

Complete the code necessary to do this. Note: You must write three lines of code.

```
// Create Dog object

Dog d = new Dog("Carmen", 7);

// Set the request attribute

request.setAttribute("dog", d);
RequestDispatcher view
     = request.getRequestDispatcher("/myDog.jsp");

// Forward to the view

view.forward(request, response);
```

10. (10 pts) Write html code (only what goes between <body> and </body>) for the web page shown below. The web page contains a form. When the button is clicked, the servlet processName handles the request.

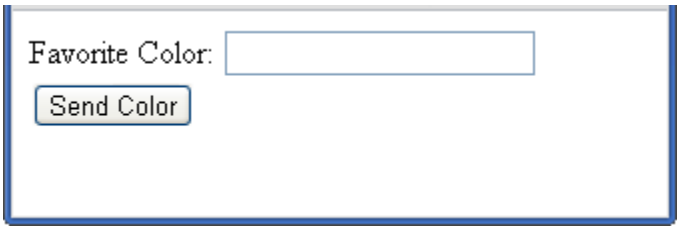



```
<body>
    <form action="processName">
        Name: <input type="text" name="userName" /><br />
        <input type="submit" value="Say Hello" />
    </form>
 </body>
```

(5 pts) Write servlet code that would assign the value of the request parameter named **`hello`** to the String variable **`greeting`**.

```
String greeting = request.getParameter("hello");
```

10. (10 pts) Write html code (only what goes between <body> and </body>) for the web page shown below. The web page contains a form. When the button is clicked, the servlet processColor handles the request.




```
<body>
    <form action="processColor">
        Favorite Color: <input type="text" name="userColor" /><br />
        <input type="submit" value="Send Color" />
```

```
    </form>
 </body>
```

11. (5 pts) Write servlet code that would assign the value of the request parameter named good**bye** to the String variable **`farewell`**.

```
String farewell = request.getParameter("goodbye");
```

AJAX

17. In a web application that uses AJAX, the input components in the user interface must be inside a <form> tag.

    A. True

    B. False

18. The AJAX response has been received from the server when request.readyState == ___4_____ and request.status == __200_____.

19. When an AJAX response is received on the client side, it is handled by a JavaScript function called the _____callback_____ function.

20. Assume we are returning an AJAX response that is written in ordinary text. In the servlet, we set the response type as follows (fill in the blank):

    response.setContentType("text/html_____;charset=UTF-8");

21. Assume we are returning an AJAX response that is written in ordinary text. On the client side, we obtain the message that was returned using the expression (fill in the blank)

request.responseText_____

22. The following jQuery code initiates an AJAX request.

```
$.get("hello", goodbye);
```

In this code, "hello" is

A. a request parameter

B. the URL of a servlet

C. the callback function

In AJAX, the request object is of type

A. Request

B. HttpRequest

C. XMLHttpRequest

20. Assume we are returning an AJAX response that is written in XML. In the servlet, we set the response type as follows (fill in the blank):

response.setContentType("text/xml_____;charset=UTF-8");

21. Assume we are returning an AJAX response that is written in XML. On the client side, we obtain the message that was returned using the expression (fill in the blank)

request.responseXML_____

22. The following jQuery code initiates an AJAX request.

```
$.get("hello", goodbye);
```

In this code, goodbye is

A. a request parameter

B. the URL of a servlet

C. the callback function

21. Assume we are returning an AJAX response that is written in JSON. In the servlet, we set the response type as follows (fill in the blank):

response.setContentType("application/json_____;charset=UTF-8");

Here is part of the code for the web page:

```
<p>
  Select Subject:<br />
  <select id="subject">
     <option value="cs">Computer Science</option>
     <option value="math">Math</option>
     <option value="physics">Physics</option>
  </select>
</p>
<p>
  <input type="   " id="getCourseList"
         value="Find Available Courses" /><br />
</p>
<ul id="courseList"></ul>
```

(a) (4 pts) Does this code have to be inside a <form> tag?

YES NO

(b) (4 pts) What value should be used for the type of the input element?

button submit

18. Here is the header for the servlet:

```
@WebServlet(name="CourseServlet",
            urlPatterns={"/courses"})
public class CourseServlet extends HttpServlet {
```

and here is the code for the action handler for the button:

```
function getCourses() {
    clearCourses();

    request = createRequest();
    if (request == null) {
        alert("Unable to create request");
        return;
    }

    var url = // See part (a)
    request.open("GET", url, true);
    request.onreadystatechange = displayCourses;
    request.send(null);
}
```

(a) (4 pts) Complete the code that assigns a value to the variable url. Be sure to include the request parameter.

```
var url = "courses?subject=" + document.getElementById("subject").value;
```

(b) (4 pts) What is the name of the callback function? displayCourses

19. (3 pts) Here is part of the code for the callback function:

```
if (request.readyState == 4) {

  if (request.status == 200) {

    var courses = // See question below

    var courseList = courses.split("\n");
    for(var i = 0; i < courseList.length - 1; ++i) {
      var item = document.createElement("li");
```

Complete the statement that assigns the information returned from the server to the variable courses. Assume the information is being returned as text.

var courses = request.responseText;

What is JSON?

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- JSON is lightweight text-data interchange format
- JSON is language independent *
- JSON is "self-describing" and easy to understand

* JSON uses JavaScript syntax for describing data objects, but JSON is still language and platform independent. JSON parsers and JSON libraries exists for many different programming languages.

JSON - Evaluates to JavaScript Objects

The JSON text format is syntactically identical to the code for creating JavaScript objects.

Because of this similarity, instead of using a parser, a JavaScript program can use the built-in eval() function and execute JSON data to produce native JavaScript objects.

Much Like XML

- JSON is plain text
- JSON is "self-describing" (human readable)
- JSON is hierarchical (values within values)
- JSON can be parsed by JavaScript
- JSON data can be transported using AJAX

Much Unlike XML

- No end tag
- Shorter
- Quicker to read and write
- Can be parsed using built-in JavaScript eval()
- Uses arrays
- No reserved words

Why JSON?

For AJAX applications, JSON is faster and easier than XML:

Using XML

- Fetch an XML document
- Use the XML DOM to loop through the document
- Extract values and store in variables

Using JSON

- Fetch a JSON string
- eval() the JSON string

JSON syntax is a subset of JavaScript syntax

23. Consider the following JavaScript code:

```
var book = { "title": "Cryptonomicon", "author": "Stephenson" };
```

Complete the following statement that assigns the title of the book to the variable bookTitle. Do NOT use the string "Cryptonomicon" in your answer.

```
var bookTitle = book.title;
```

24. Suppose we use the jQuery function $.each() to iterate over an array. The current item in the array is referred to using the name

A. index

B. this

C. element

25. Assume the Java variable gen of type JsonGenerator. To start writing a JSON object, we call gen._____().

A. write

B. start

C. writeStartObject

24. To iterate over an array, we can use the jQuery function

A. for

B. each

C. foreach

25. Assume the Java variable gen of type JsonGenerator. To write a name-value pair for an attribute of a JSON object, we call gen._____().

A. write

B. attribute

C. writeAttribute

25. Assume the Java variable gen of type JsonGenerator. To finish writing a JSON object, we call gen._____().

A. end

B. writeEnd

C. writeEndObject

JavaServer Pages

- Welcome page: forms, request parameters, submit button
- Controller servlet: access request parameters, create request attributes, forward to view
- Response page: access request parameters and attributes using
- EL
- JSTL tags: if, choose, forEach

JavaServer Faces

- Static and dynamic page navigation: examples 2 and 3
- Internationalization, resource bundles, locales: example 4
- Data tables: example 5.
- Validators and converters: example 6
- Facelet templates: Guess number example.

JSP

1. Ordinary html can be included in a JavaServer Pages page.

A. True

B. False

2. JSTL Tags can be included in a JavaServer Pages page.
   A. True
   B. False

3. In MVC, the letter M stands for
   A. Mobile
   B. Model
   C. Master

4. In MVC, the letter V stands for
   A. Vision
   B. Version
   C. View

5. In MVC, the letter C stands for
   A. Controller
   B. Computing
   C. Command

6. Expressions written in JSP Expression Language are evaluated
   A. On the client side.
   B. On the server side.

7. Assume Drink is a Java class that has two attributes: type and size. A servlet executes this code:

```
Drink drinkOrder = new Drink("Tea", 16);
```

Write Java code that makes this object the value of a request attribute named "order".

```
request.setAttribute("order", drinkOrder);
```

8. The servlet in Problem 7 then forwards to ConfirmOrder.jsp. Write code that could be included in ConfirmOrder.jsp to display the size of drink the customer ordered. You must use an EL expression to obtain the size of the drink.

```
${order.size}
```

Questions 9-12 pertain to this code for a start page:

```
<body>
   <_____________________   _____________________="Response.jsp">
      Ice cream flavor:
      <input type="text" name="flavor">
      <input type="_____________________" value="Order">
   </_____________________>
```

```
</body>
```

9. What is the missing tag in the second and sixth lines? ___form_____

10. What is the missing attribute in the second line? __action_____

11. What is the missing attribute value in the fifth line? __submit_____

12. Write an EL expression that could be used in Response.jsp to display the ice cream flavor the user entered.
    `${param.flavor}`

Assume the value of param.flavor is a flavor of ice cream. Write code to be included in a JavaServer Page to display "Good Choice" if the flavor is "chocolate" and "Bad Choice" for all other flavors. Use the JSTL <c:choose> tag to write your code.

```
<c:choose>
   <c:when test="${param.flavor eq 'chocolate'}">
      Good Choice
   </c:when>
   <c:otherwise>
      Bad Choice
   </c:otherwise>
</c:choose>
```

13. Assume mouseList is an array of Strings. Each entry in the array is the name of a mouse. A servlet sets a request attribute named "mice" with value mouseList. It then forwards the request to a JavaServer Page. Complete the code below that would be included in a JavaServer Page to display the list of mouse names in an HTML table.

```
<table>
   <c:__forEach_____________ var="mouse" items="${mice}">
      <tr>
         <td>___${mouse}_____________</td>
      </tr>
   </c:__forEach_____________>
</table>
```

14. Expressions written in JSP Expression Language are evaluated
    A. On the server side
    B. On the client side.

15. Assume the value of param.cost is the cost of a Christmas present. Write code to be included in a JSP to display

- OK if the cost is ≤ 100
- NOT OK if the cost is > 100

Use the JSTL <c:if> tag to write your code.

```
<c:if test="${param.cost le 100}">
   NOT
</c:if>
OK
```

16. Assume dogList is an array of Strings. Each entry in the array is the name of a dog. A servlet sets a request attribute named "dogs" with value dogList. It then forwards the request to a JavaServer Page. Complete the code below that would be included in a JavaServer Page to display the list of dog names in an HTML table.

```
<table>
   <c:forEach ___var________="d" __items________="${dogs}">
      <tr>
         <td>${d}</td>
      </tr>
   </c:forEach>
</table>
```

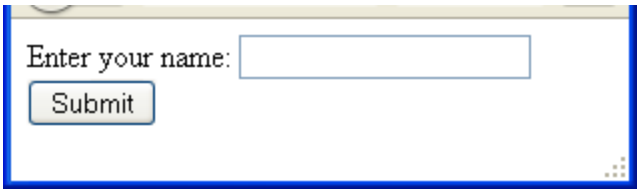
17. Assume the value of param.cost is the cost of a Christmas present. Write code to be included in a JSP to display

- CHEAP if the cost is < 10
- NOT CHEAP if the cost is >= 10

Use the JSTL <c:if> tag to write your code.

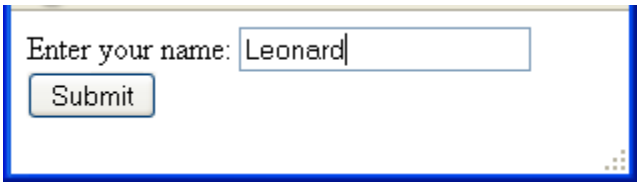
```
<c:if test="${param.cost ge 10}">
   NOT
</c:if>
CHEAP
```

10 pts) Write the code for the body of the welcome page shown below. When the button is clicked, the response page "GreetUser.jsp" is returned. Write only the code that goes inside the <body> tag of the welcome page.

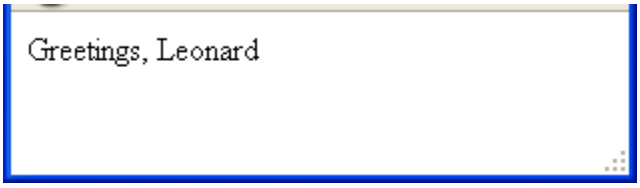



```
<form action="GreetUser.jsp">
    Enter your name: <input type="text" name="userName" /><br />
    <input type="submit" value="Submit" />
</form>
```

2. (7 pts) (Continuation of Problem 1.) The response page displays "Greetings, " followed by the user name that was entered in the welcome page. For example, if the user types "Leonard" in the input box,



the response page looks like this:



Write the code for the body of the response page. You MUST use EL to obtain the user name. Write only the code that goes inside the <body> tag of the response page.

```
Greetings, ${param.userName}
```

3. (9 pts) Assume class **Movie** has the constructor shown below:

```
public Movie(String title, int rating)
```

In an MVC application, the controller

- creates a **Movie** object with title "The Matrix" and rating 4,
- makes the object the value of a request attribute named **scifiMovie**
- forwards to the view

Complete the code necessary to do this. **Note: You must write three lines of code.**

```
// Create Movie object

Movie matrix = new Movie("The Matrix", 4);
```

```
// Set the request attribute

request.setAttribute("scifiMovie", matrix);
```

```
RequestDispatcher view

    = request.getRequestDispatcher("/moviePage.jsp");
```

```
// Forward to the view

view.forward(request, response);
```

4. (6 pts) (Continuation of Problem 3.) Write the code that would be included in the response page moviePage.jsp to display the following:

Title: The Matrix

Rating: 4

Write only the code necessary to display the text shown above. **You must use EL expressions to obtain the title and the rating of the movie**.

```
Title: ${scifiMovie.title}<br />

Rating: ${scifiMovie.rating}
```

5. (8 pts) Assume the value of param.title is the title of a movie. Write code to be included in a JSP to display

- Dream if the title is Inception
- Lizard if the title is Rango
- Boring for all other titles

Use the JSTL <c:choose> tag to write your code.

```
<c:choose>
   <c:when test="${param.title eq 'Inception'}">
      Dream
   </c:when>
   <c:when test="${param.title eq 'Rango'}">
```

```
      Lizard
   </c:when>
   <c:otherwise>
      Boring
   </c:otherwise>
</c:choose>
```

6. (10 pts) Assume bookList is an array of Strings. Each entry in the array is the title of a book. A servlet executes the statement below and then forwards the request to a JavaServer Page.

```
request.setAttribute("books", bookList);
```

Complete the code below that would be included in a JavaServer Page to display the list of book titles in an **HTML table**. Use a JSTL <c:forEach> tag.

```
<h2>Book List:</h2><br>
<table>

<c:forEach var="title" items="${books}">
      <tr>
         <td>${title}</td>
      </tr>
   </c:forEach>
</table>
```

7. (5 pts) Assume we have a request parameter named "dogName". Write an EL expression that could be included in a JSP to obtain the dog's name.

```
${param.dogName}
```

8. (6 pts) The class Car is defined as shown below:

```
public class Car
{
  private String make;
  private String model;

  public Car(String make, String model)
  {
    this.make = make;
    this.model = model;
  }

  public String getMake()
  {
    return make;
  }

  public String getModel()
  {
    return model;
  }
}
```

A servlet executes the following code:

```
Car theCar = new Car("Toyota", "Prius");
request.setAttribute("car", theCar);
RequestDispatcher view
  = request.getRequestDispatcher("/myToy.jsp");
view.forward(request, response);
```

Write code that could be included in a JSP to display the car's make and model. **You must use EL to do this**.

```
The make of the car is ${car.make}
```

```
The model of the car is ${car.model}
```

9. (5 pts) Assume a welcome page contains this form:

```
<form action="SayHi.jsp" method="post">
  Movie: <input type="text" name="movieName"/><br>
  <input type="submit" value="Submit" />
</form>
```

If the user enters **Armagedon** for the movie, the response must be

Armagedon

Hello, asteroid

Otherwise, the response is only the name of the movie.

Write code that would be included in **SayHi.jsp** to produce this response. You must use the JSTL **<c:if>** tag and EL. Do NOT write the complete .jsp file—just show the code necessary to produce the response described above.

```
${param.movieName}<br>
<c:if test="${param.movieName eq 'Armagedon'}">
   Hello, asteroid
</c:if>
```

10. (5 pts) Assume the value of **`param.movie`** is the name of a movie. Write code to be included in a .jsp file to display

- **`Neytiri`** if the movie is **`Avatar`**,
- **`Neo`** if the movie is **`The Matrix`**, and
- **`Arnold`** for all other movies.

Use the JSTL **`<c:choose>`** tag to write your code.

```
<c:choose>
    <c:when test="${param.movie eq 'Avatar'}">
        Neytiri
    </c:when>
    <c:when test="${param.movie eq 'The Matrix'}">
        Neo
    </c:when>
    <c:otherwise>
        Arnold
    </c:otherwise>
</c:choose>
```

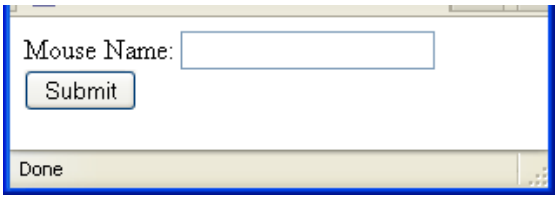
11. (5 pts) Assume **movieList** is an array of Strings. Each entry in the array is a movie title. A servlet executes the statement below and then forwards the request to a Java Server Page.

```
request.setAttribute("movies", movieList);
```

Complete the code below that would be included in the Java Server Page to display the list of movie titles in an unordered list. Use a JSTL **<c:forEach>** tag.

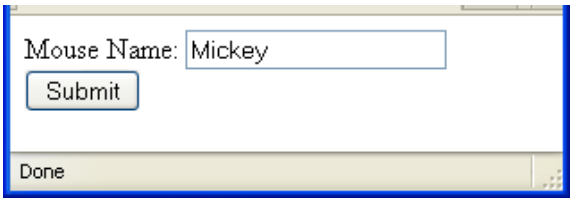
```
<h2>Movie List:</h2><br>
<ul>
  <c:forEach var="movie" items="${movies}">
      <li>${movie}<li>
   </c:forEach>
</ul>
```

1. (12 pts) Write the code for the body of the welcome page shown below. When the button is clicked, the response page "Hello.jsp" is returned. Write only the code that goes inside the <body> tag of the welcome page.

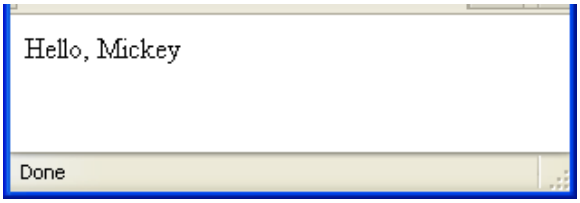


```
<form action="Hello.jsp">
   Mouse Name: <input type="text" name="mouseName" /><br />
   <input type="submit" value="Submit" />
</form>
```

2. (10 pts) (Continuation of Problem 1.) The response page displays "Hello, " followed by the mouse name that was entered in the welcome page. For example, if the user types "Mickey" in the input box,



the response page looks like this:



Write the code for the body of the response page. You MUST use EL to obtain the mouse name. Write only the code that goes inside the <body> tag of the response page.

```
Hello, ${param.mouseName}
```

3. (9 pts) Assume class **`Student`** has the constructor shown below:

```
public Student(String name, int id)
```

In an MVC application, the controller

- creates a **`Student`** object with name "Carmen" and id 919191919,
- makes the object the value of a request attribute named **`honorStudent`**
- forwards to the view

Complete the code necessary to do this. **Note: You must write three lines of code.**

```
// Create Student object

Student carmen = new Student("Carmen", 919191919);
```

```
// Set the request attribute

request.setAttribute("honorStudent", carmen);
```

```
RequestDispatcher view

     = request.getRequestDispatcher("/studentPage.jsp");
```

```
// Forward to the view

view.forward(request, response);
```

4. (6 pts) The class Canine is defined as shown below:

```
public class Canine
{
  private String name;
  private String major;

  public Canine(String name, String major)
  {
    this.name = name;
    this.major = major;
  }

  public String getName()
  {
    return name;
  }

  public String getMajor()
  {
    return major;
  }
}
```

A servlet executes the following code:

```
Canine smartDog = new Canine("Leonard", "Math");
```

```
request.setAttribute("poodle", smartDog);
```

and then forwards to a JSP. **Write an EL expression** that could be included in the JSP to display the canine's major.

```
${poodle.major}
```

5. (12 pts) Assume the value of param.creature is the name of an animal. Write code to be included in a JSP to display

- Dog if the name is Leonard
- Cat if the name is Maude
- Mouse for all other names

Use the JSTL <c:choose> tag to write your code.

```
<c:choose>
   <c:when test="${param.creature eq 'Leonard'}">
      Dog
   </c:when>
   <c:when test="${param.creature eq 'Maude'}">
      Cat
   </c:when>
   <c:otherwise>
      Mouse
   </c:otherwise>
</c:choose>
```

6. (12 pts) Assume catList is an array of Strings. Each entry in the array is the name of a cat. A servlet executes the statement below and then forwards the request to a JavaServer Page.

```
request.setAttribute("cats", catList);
```

Complete the code below that would be included in a JavaServer Page to display the list of cat names in an **HTML table**. Use a JSTL <c:forEach> tag.

```
<h2>Cat List:</h2><br>
<table>
   <c:forEach var="cat" items="${cats}">
      <tr>
         <td>${cat}</td>
      </tr>
   </c:forEach>
</table>
```

JSF

18. JavaServer Faces is a _____ technology.
    A. client side
    B. server side

19. JavaServer Faces pages can include ordinary html code.
    A. True
    B. False

20. A language-country combination is called a
    A. converter
    B. locale
    C. property

21. The welcome page index.xhtml for a JSF application contains the following code for a button inside a form:

```
<h:commandButton value="Quit"     action="Goodbye"      />
```

Complete the code so that the application navigates to the page Goodbye.xhtml when the button is clicked.

22. Suppose we want to internationalize a web application so that it can display text in either English or French. We have declared a resource bundle in faces-config.xml:

```
<application>
   <resource-bundle>
      <base-name>
         MyText
      </base-name>
      <var>labels</var>
   </resource-bundle>
</application>
<application>
   <locale-config>
      <default-locale>
         en_US
      </default-locale>
      <supported-locale>
```

```
            fr_FR
        </supported-locale>
    </locale-config>
</application>
```

(a) We need a label **student** with text "Student name" (or " Nom de l'étudiant" in French).

Write the entry that must be included in the file MyText.properties.

```
student=Student name
```

Write the entry that must be included in the file MyText_fr_FR.properties.

```
student=Nom de l'étudiant
```

(b) Complete the JSP code below that displays the label.

```
<h:outputText value="#{labels.student}" />
```

20. The code below is used to input a cat's age in a text box.

```
<h:inputText
   value="#{UserBean.catAge}"
   id="ageField"
/>
```

Rewrite the code so that the user is required to enter a value for this field.

```
<h:inputText
   value="#{UserBean.catAge}"
   id="ageField"
   required="true"
/>
```

21. Rewrite the original code in Problem 20 to validate that the user enters a value between 0 and 25 (inclusive).

```
<h:inputText value="#{UserBean.catAge}"
             id="ageField">
   <f:validateLongRange
```

```
        minimum="0"
        maximum="25" />
</h:inputText>
```

22. Rewrite the code below by adding a converter so that totalOrder will be displayed as currency.

```
<h:outputText value="#{shoppingCartBean.totalOrder}">

   <f:convertNumber type="currency" />

</h:outputText>
```

23. One of the examples in the JSF unit displayed a grade report for students in a class. It used a dataTable element to display the report. The beginning of the code for the dataTable is shown below. Write the code to create a column in the dataTable that displays the final exam grades of the students. The column heading must be "Final Exam". Note: The Student class has a method getFinalExam().

```
<h:dataTable value="#{gradeBook.students}" var="student">

   <h:column>
      <f:facet name="header">Final Exam</f:facet>
      #{student.finalExam}
```

```
</h:column>
```

24. Here is some code from a JSF template:

```
<h:body>
   <ui:insert name="cat"></ui:insert>
   <br />
   <ui:insert name="dog"></ui:insert>
</h:body>
```

Complete the code in the facelet below so that the web page produced looks like the following when displayed in a browser.


```
<h:body>
   <ui:composition template="./template.xhtml">

      <ui:define name="cat">
         Fluffy
      </ui:define>

      <ui:define name="dog">
         Fang
      </ui:define>
```

```
   </ui:composition>
</h:body>
```

25. The code below appears inside a form element on a JSF page.

```
<h:inputText value="#{UserBean.salary}" />
```

What method is invoked by the expression `#{UserBean.salary}` when the web page is displayed?

A. setSalary()

B. getSalary()

C. salary()

(4 pts) The welcome page index.xhtml for a JSF application contains the following code for a button inside a form:

```
<h:commandButton value="Click Me"  action="CheckOut"     />
```

Complete the code so that the application navigates to the page CheckOut.xhtml when the button is clicked.

8. (4 pts) The code below appears inside a form element on a JSF page.

```
<h:inputText value="#{UserBean.studentGpa}" />
```

(a) What method is invoked by the expression #{UserBean.studentGpa} when the web page is displayed?

A. studentGpa()

B. getStudentGpa()

C. setStudentGpa()

(b) What method is invoked by the expression #{UserBean.studentGpa} when the form is submitted?

A. studentGpa()

B. getStudentGpa()

C. setStudentGpa()

9. (6 pts) Suppose we want to internationalize a web application so that it can display text in either English or French. We have declared a resource bundle in faces-config.xml:

```
<application>
   <resource-bundle>
      <base-name>
         MyText
      </base-name>
      <var>txt</var>
   </resource-bundle>
</application>
<application>
   <locale-config>
      <default-locale>
         en_US
      </default-locale>
      <supported-locale>
         fr_FR
      </supported-locale>
   </locale-config>
</application>
```

(a) We need a label **title** with text "Movie title" (or "Titre du film" in French).

Write the entry that must be included in the file MyText.properties.

```
title=Movie title
```

Write the entry that must be included in the file MyText_fr_FR.properties.

```
title = Titre du film
```

(b) Complete the JSF code below that displays the label.

```
<h:outputText value="#{txt.title}"
```

10. (4 pts) The code below is used to input a movie's rating in a text box.

```
<h:inputText
   value="#{movieBean.movieRating}"
   id="ratingField"
/>
```

Rewrite the code so that the user is required to enter a value for this field.

```
<h:inputText
   value="#{movieBean.movieRating}"
   id="ratingField"
   required="true"
/>
```

11. (10 pts) Rewrite the original code in Problem 10 to validate that the user enters a value between 1 and 5 (inclusive).

```
<h:inputText
      value="#{movieBean.movieRating}"
      id="ratingField">
   <f:validateLongRange
      minimum="1"
      maximum="5" />
</h:inputText>
```

12. (6 pts) Rewrite the code below by adding a converter so that priceOfTicket will be displayed as currency.

```
<h:outputText value="#{movieBean.priceOfTicket}">

    <f:convertNumber type="currency" />

</h:outputText>
```

13. (8 pts) One of the examples in the JSF unit displayed a grade report for students in a class. It used a dataTable element to display the report. The beginning of the code for the dataTable is shown below. Write the code to create a column in the dataTable that displays the names of the students. The column heading must be "Name".

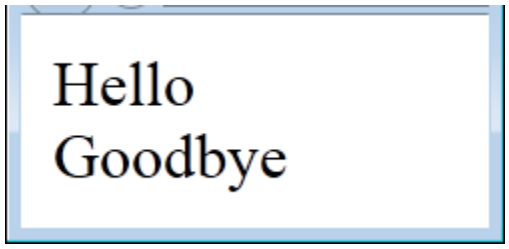
```
<h:dataTable value="#{gradeBook.students}" var="student">

    <h:column>
        <f:facet name="header">Name</f:facet>
        #{student.name}
```

```
</h:column>
```

14. (8 pts) Here is some code from a JSF template:

```
<h:body>
   <ui:insert name="greeting"></ui:insert>
   <br />
   <ui:insert name="salutation"></ui:insert>
</h:body>
```

Complete the code in the facelet below so that the web page produced looks like the following when displayed in a browser.




```
<h:body>
   <ui:composition template="./template.xhtml">

      <ui:define name="greeting">
         Hello
      </ui:define>

      <ui:define name="salutation">
         Goodbye
      </ui:define>
```

```
        </ui:composition>
</h:body>
```

Here are some tags you might need to use in Part 1 of this exam.

```
<h:column>
<default-locale>
<f:facet>
<from-outcome>
<from-view-id>
<managed-bean>
<managed-bean-class>
<managed-bean-name>
<managed-bean-scope>
<h:message>
<navigation-case>
<navigation-rule>
<h:outputText>
<resource-bundle>
<supported-locale>
<to-view-id>
<f:validateLongRange>
```

1. (10 pts) A navigation rule does the following: If the user is on the page **dog.jsp** and takes an action producing an outcome of **speak**, then the page **woof.jsp** should be displayed. If the user takes an action producing an outcome of **stay**, then the page **sit.jsp** should be displayed. Write the navigation rule that must be included in faces-config.xml. It is not necessary for you to include a description.

```
<navigation-rule>
    <from-view-id>/dog.jsp</from-view-id>
    <navigation-case>
        <from-outcome>speak</from-outcome>
        <to-view-id>/woof.jsp</to-view-id>
    </navigation-case>
    <navigation-case>
        <from-outcome>stay</from-outcome>
        <to-view-id>/sit.jsp</to-view-id>
    </navigation-case>
</navigation-rule>
```

2. (2 pts) In the code below

```
<h:outputText
   value="#{UserBean.studentGPA}"
/>
```

The expression #{UserBean.studentGPA} invokes what method?

A. studentGPA()

B. getStudentGPA ()

C. setStudentGPA ()

3. (3 pts) Rewrite the code below so that the user is <u>required</u> to enter a value for this field. You do NOT need to write code to display an error message.

```
<h:inputText id="payments"
             value="#{CarBean.numberOfPayments}"
             required="true">
</h:inputText>
```

4. (4 pts) Rewrite the code in Problem 3 to validate that the user enters a value between 12 and 48 (inclusive). You do NOT need to write code to display an error message.

```
<h:inputText id="payments"
             value="#{CarBean.numberOfPayments}">
   <f:validateLongRange
       minimum="12"
       maximum="48" />
</h:inputText>
```

5. (10 pts) Assume we want to use a managed bean in a JSF application. The bean has the name **Pinto**, it is defined in the file **Legume.java**, and it has **request** scope. Write the code that must be included in faces-config.xml.

```
<managed-bean>
    <managed-bean-name>Pinto</managed-bean-name>
    <managed-bean-class>Legume</managed-bean-class>
    <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
```

6. (4 pts) Suppose we want to internationalize a web application so that it can display text in either English or French. We have declared a resource bundle in faces-config.xml:

```
<application>
    <resource-bundle>
        <base-name>
            studentExam.MyText
        </base-name>
        <var>examLabels</var>
    </resource-bundle>
</application>
<application>
    <locale-config>
        <default-locale>
            en_US
        </default-locale>
        <supported-locale>
            fr_FR
        </supported-locale>
    </locale-config>
</application>
```

(a) We need a label **examScore** with text "Your score must be at least eighty."

("Votre score doit être d'au moins quatre-vingts." in French).

Write the entry that must be included in the file studentExam.MyText.properties.

examScore=Your score must be at least eighty.

Write the entry that must be included in the file studentExam.MyText_fr_FR.properties.

examScore= Votre score doit être d'au moins quatre-vingts.

(b) Complete the JSP code below that displays the label.

```
<h:outputText value="#{examLabels.examScore}"/>
```

7. (10 pts) Assume

- a Computer object has getNumberOfCores() and getMemorySize() methods
- the expression #{Inventory.computers} returns a list of Computer objects.

Complete the code shown below to produce a table listing the number of cores and memory size of each of the computers. The headings for the columns must be "Cores" and "Memory".

```
<h:dataTable value="#{Inventory.computers}"
             var="computer"
             border="1"
             style="border:solid 1px">
    <h:column>
       <f:facet name="header">
          <h:outputText value="Cores"/>
       </f:facet>
       <h:outputText value="#{computer.numberOfCores}"/>
    </h:column>
    <h:column>
       <f:facet name="header">
          <h:outputText value="Memory"/>
       </f:facet>
       <h:outputText value="#{computer.memorySize}"/>
    </h:column>

</h:dataTable>
```

(5 pts) The welcome page index.xhtml for a JSF application contains the following code for a button inside a form:

<h:commandButton value="Click Me" action="Hi" />

Complete the code so that the application navigates to the page Hi.xhtml when the button is clicked.

8. (5 pts) In the code below

<h:inputText value="#{UserBean.studentGpa}" />

the expression #{UserBean.studentGpa} invokes what method?

A. studentGpa()

B. getStudentGpa()

C. setStudentGpa()

9. (7 pts) Suppose we want to internationalize a web application so that it can display text in either English or French. We have declared a resource bundle in faces-config.xml:

```
<application>
    <resource-bundle>
        <base-name>
            MyText
        </base-name>
        <var>labels</var>
    </resource-bundle>
</application>
<application>
    <locale-config>
        <default-locale>
            en_US
        </default-locale>
        <supported-locale>
            fr_FR
        </supported-locale>
    </locale-config>
</application>
```

(a) We need a label **student** with text "Student name" (or " Nom de l'étudiant" in French).

Write the entry that must be included in the file MyText.properties.

```
student=Student name
```

Write the entry that must be included in the file MyText_fr_FR.properties.

```
student=Nom de l'étudiant
```

(b) Complete the JSP code below that displays the label.

```
<h:outputText value="#{labels.student}" />
```

10. (5 pts) The code below is used to input a dog's age in a text box.

```
<h:inputText
   value="#{UserBean.dogAge}"
   id="ageField"
/>
```

Rewrite the code so that the user is required to enter a value for this field.

```
<h:inputText
    value="#{UserBean.dogAge}"
    id="ageField"
    required="true"
/>
```

11. (10 pts) Rewrite the original code in Problem 10 to validate that the user enters a value between 2 and 18 (inclusive).

```
<h:inputText value="#{UserBean.dogAge}"
             id="ageField">
   <f:validateLongRange
       minimum="2"
       maximum="18" />
</h:inputText>
```

12. (7 pts) Rewrite the code below by adding a converter so that totalCost will be displayed as currency.

```
<h:outputText value="#{carBean.totalCost}">

  <f:convertNumber type="currency" />

</h:outputText>
```