CSE135 Winter 05, Final Exam

Student Name:

Student ID:

There are 160 points. However, the top score will be 140 points. This means that losing 20 points here and there will not matter to your eventual grade. Nevertheless, go for 160 points, since you may lose points in a few answers.

You have 180 minutes. When you finish, leave the exam on your desk.

Good Luck!
Resource Access (4 points)
Choose the correct answer (no justification needed). Servlets typically have access to
1. Database resources on the web server side only
2. Database resources on the web client side only
3. Database resources on both the web server and the web client side

Tomcat and Static Content (6 points)
Provide a yes or no answer to the following and add an appropriate short justification:
If you use Tomcat as your application server (i.e., servlet container, jsp processor) then the static content (html pages) must also be served by Tomcat.
1. Yes, because …
2. No, because …

Customization using the web.xml (12 points)
You are a consultant and just finished a web application for ABC Inc. ABC wants every page of the web application to have on the bottom line the message:

“For complaints and troubleshooting please email joe@abc.com”

However, every few months ABC needs to change the recipient of complaint calls. The ABC employees would hate to have to modify your jsp and java beans code in order to effect the change. They are also clueless about databases. Produce a solution where they only have to make some change in the web.xml; i.e., have the name of the recipient in the web.xml. Show what should be included in the web.xml and also show the code that you will include in the jsp to produce the line shown above (ignore typesetting).
Access to Application-Scope Attributes (12 points)

Consider the following modification of the counter application, where two counters \( A \) and \( B \) are incremented every time one visits the jsp page. Is it possible that a user sees on his/her browser:

“A=2, B=3”

If no, outline why. If yes, provide a sequence of events that leads to this situation.

<HTML>
<HEAD>
  <TITLE>Counter Web Application</TITLE>
</HEAD>
<BODY>
<% Integer A =
(Integer)(application.getAttribute("counter")) ;
   if (A == null) { A = new Integer(0) ; }
   else { A = new Integer(A.intValue() + 1) ; }
   application.setAttribute("counter", A) ; %>
<!--some static content is produced here-->
<% Integer B =
(Integer)(application.getAttribute("counter")) ;
   if (B == null) { B = new Integer(0) ; }
   else { B = new Integer(B.intValue() + 1) ; }
   application.setAttribute("counter", B) ; %>
A=<%= A %>, B=<%= B %>. 
</BODY>
</HTML>
Reminding Potential Customers What They Wanted (30 points)
Consider an e-commerce application, built using jsp, where potential customers browse for CDs. The `browsing.jsp` displays CDs and next to each CD’s data there is a “put in shopping cart” hyperlink. For example, for the CD “XYZ” it is

```html
<a href="browsing.jsp?action=buy&cd=XYZ">Put XYZ in the shopping cart</a>
```

The `browsing.jsp` has the following code that inserts the CD on the cart.

```jsp
<% String action=request.getParameter("action") ;
   if (action!=null && action.equals("buy"))
   {
   String CD=request.getParameter("cd") ;
   // here goes code that inserts the cd in
   // a session-scoped “cart” data structure
   }
%>
```

The application has no “logging in” feature; the customers can browse and build a cart without declaring who they are. It turns out that many potential customers build a cart but do not purchase it, perhaps because they are not sufficiently enthusiastic when they see the total price. However prices change and many items often go on sale. The business executives believe that sales would be much better if the `browsing.jsp` could display on the top of the page lists of CDs such as the following

```html
<p>
In the past you were interested in the following CDs that are now on sale:
<ul>
   <li>Thriller</li>
   <li>Nevermind</li>
</ul>
</p>
```

This requires that the application can figure out that in the past the potential customer had put the corresponding CDs in his/her cart but eventually did not buy them. Assume there is a boolean function `model.isItOnSale(String CdName)` that given a CD name it returns true if the CD is on sale.

- Write a piece of code that creates the required list of CDs – such as the example list above.
- Also extend the piece of code at the top of the page to achieve the purpose.

You may use cookies. Write actual Java code. It is OK to include in the on-sale list a CD that the potential customer has already put in his new shopping cart.
Send to Google (10 points)
Consider an html page yourpage.html that has the following form, which issues a request for the yourjsp.jsp.

<form method="get" action="yourjsp.jsp">

You do some processing at yourjsp.jsp and then send the user to Google’s home page. Show how you can do this at yourjsp.jsp without using html meta tags.

Struts (10 points)
In Struts you often pass an object from your Action bean to a jsp page. Describe two ways to achieve this purpose. Comment on which one is better.
Exceptions (10 points)
Your JSP page includes the following fragment

<% try{
    ....
}catch(exception ex){
    ...
}
%>

In the catch block you can redirect the user to an html page or to another jsp page. Why sending to a JSP page is typically better than an html page.
**Struts Application Design (66 points)**

Consider an e-commerce application that provides a page that looks like this and allows the customer to say, for example, “I want to include in the shopping cart $3000 worth of steel”.

\[\text{Our Web Site Allows you To Buy Anything You May Think Of!} \]

I want to include in the cart $3000 worth of steel

PUT IN CART

PROCEED TO CHECK-OUT

Upon clicking the “PUT IN CART” button the business logic of the application first validates that the $ amount is a reasonable number (assume that the check is done by a given boolean function `model.validateDollarAmount(String num)`) and the product is a reasonable string (assume that the check is done by a given boolean function `model.validateProduct(String product)`). If the dollar amount does not pass validation then the customer should be send back to the same page, where on the top there will be a message “Re-enter another $ amount”. If the product name does not pass validation then the customer should be send back to the same page, where on the top there will be a message “Re-enter product name”. The forms should be precompleted with the data previously entered. See here an example:
Our Web Site Allows you To Buy Anything You May Think Of!

I want to include in the cart $\text{gazillion}@#$%^ worth of

PUT IN CART

PROCEED TO CHECK-OUT

Once the form data pass validation the business logic uses a given boolean function `model.illegalProduct(String productName)` to decide whether the product the customer requested is legal or illegal.

- If the product is legal then the business logic invokes a given function `model.insertInCart(String amount, String productName)` and sends the customer back to Page A, with forms being clear.
- If the product is illegal then the business logic invalidates the session and sends the customer to the following html page B.
We do not want your business. Get lost now.

If the customer clicks the “PROCEED TO CHECK OUT” button of Page A then the customer is sent to the following page (the product and amount in the forms are ignored).

Provide Credit Card Number

1111 2222 3333 4444

SUBMIT
Upon submitting the business logic invokes a boolean function
model.buyTheCart(String CreditCard), which if it succeeds the customer is
sent back to Page A and otherwise he is sent back to Page C.

Write a struts program that implements the above. You may use any of the given model
functions above. Provide the following pieces:

- Html code of pages A and C
- struts-config.xml (we provide a template in the next page so that you
do not write tags by hand)
- form beans, including validation when needed
- action beans

Ignore the following:
- Ignore logging aspects (i.e., how does the business logic knows who is the
customer currently using the application).
- Ignore validation of the credit card number. Assume that the given credit card is
definitely valid, i.e., do not validate the credit card number.
- You do not have to write the html code of Page B ☺
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE ...ignore...>
<struts-config>
<!-- =====================Form Bean Definitions -->
<!-- You may have more or less than the 3 form beans
for which we have made space -->
<form-beans>
  <form-bean
    name=
    type= />
  <form-bean
    name=
    type= />
  <form-bean
    name=
    type= />
</form-beans>

<!-- ================== Global Forward Definitions -->
<global-forwards>
  <forward
    name=
    path= />
  <forward
    name=
    path= />
  <forward
    name=
    path= />
</global-forwards>

<!-- ================  Action Mapping Definitions -->
&action-mappings>
  <action
    path=
    type=
    parameter= />
  <action
    path=
    type=
    name= /
<struts-config>

<action-mappings>

  <forward name=" " />
  <forward name=" " />

</action-mappings>

</struts-config>