CS490bk Server-side Web Applications
Spring/Summer, 2018

Assignment 1

Assigned Date: Wednesday, June 13, 2018
Due Date: Midnight Wednesday, June 27, 2018

Please read the marking scheme and course outline before you do the assignment. For this assignment you must submit the project folders and screen shots output in a zip file. There should be one project folder for each question.

Question 1 (20 Marks)

Create a sample online order form as shown in Figure 1 to let user enter details of order. Write a servlet called ReadParamServlet to read all the form parameters entered by user. After user clicking the Submit Order button, the ReadParamServlet will display the order details in a table as shown in Figure 3 below as response, assuming the details entered are as shown in Figure 2.

If user forgot to enter a value, your ReadParamServlet should print no value in the parameter value column as shown in Figure 4. In this example user forgot to enter the address.

⚠ Problem loading page   ✕ | 👁 Applications   ✕ | An Order Form   ✕ | +

← ⓘ | localhost:8080/ReadAllParam/

Order Form

Item Number: _____
Description: _____
Price Each: $_____

First Name: _____
Last Name: _____
Middle Initial: _____

Shipping Address: [   ]
Credit Card:
   ○ Visa
   ○ MasterCard
   ○ American Express
   ○ Discover
   ○ Java SmartCard
Credit Card Number: _____
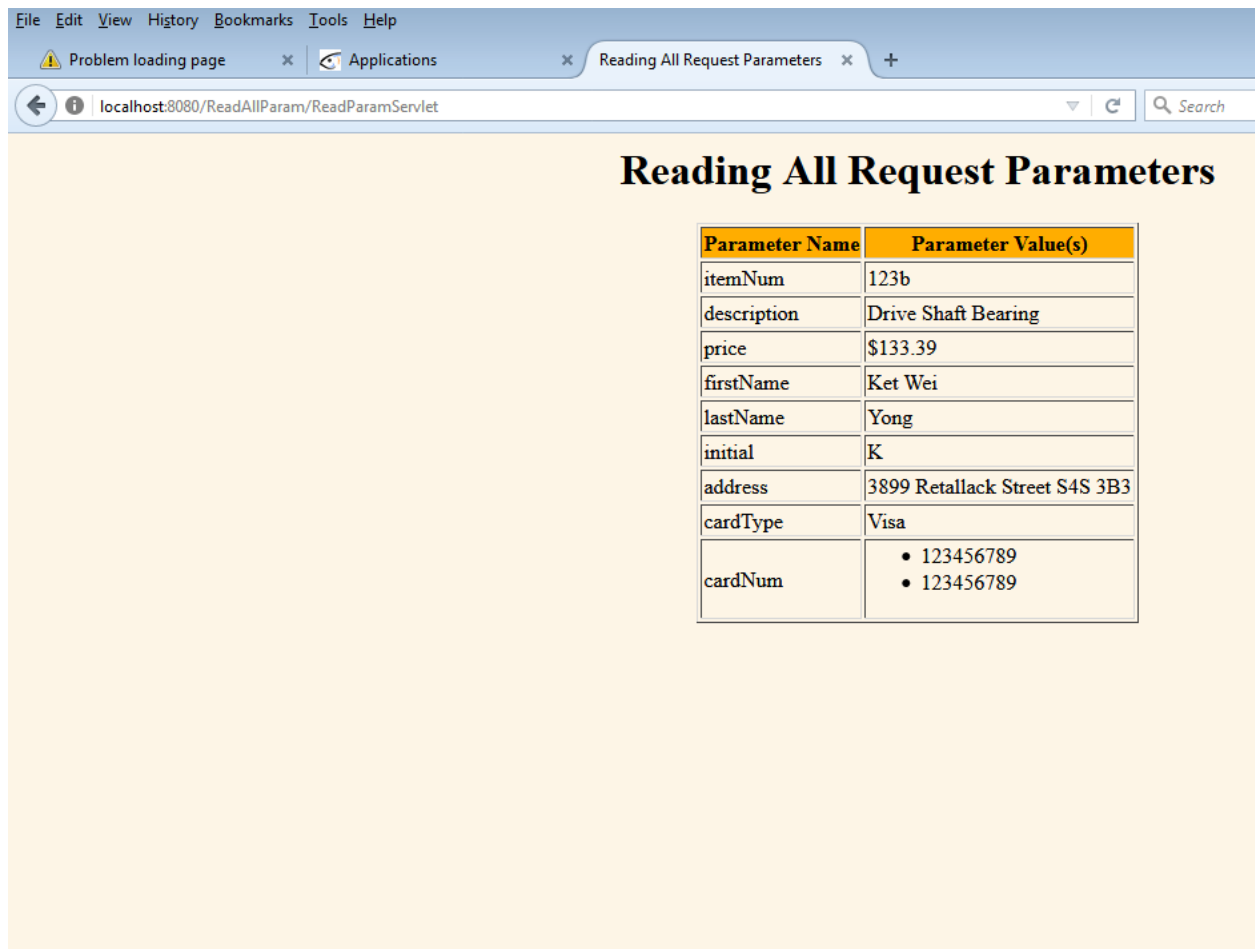Repeat Credit Card Number: _____

[Submit Order]

Figure 1

Order Form

Item Number: `123b`
Description: `Drive Shaft Bearing`
Price Each: `$133.39`

First Name: `Ket Wei`
Last Name: `Yong`
Middle Initial: `K`

```
3899 Retallack Street
S4S 3B3
```

Shipping Address:
Credit Card:
- ● Visa
- ○ MasterCard
- ○ American Express
- ○ Discover
- ○ Java SmartCard

Credit Card Number: `••••••••`
Repeat Credit Card Number: `••••••••`

[Submit Order]

Figure 2
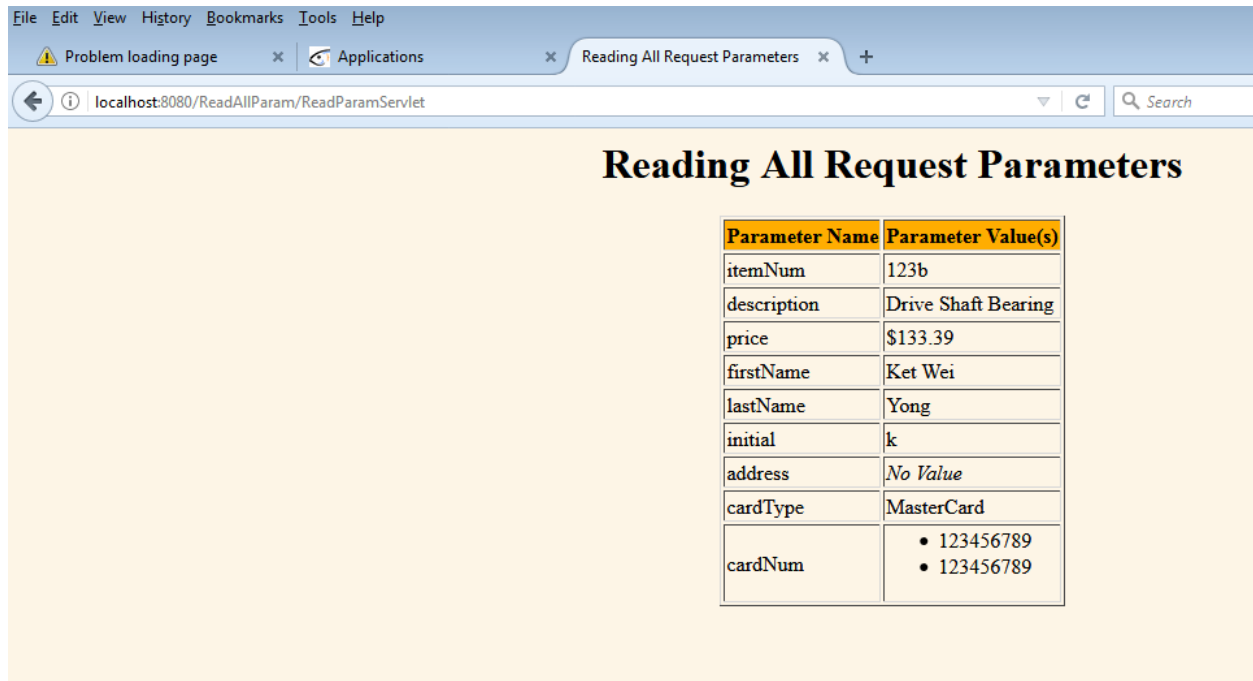
Reading All Request Parameters

Parameter Name	Parameter Value(s)
itemNum	123b
description	Drive Shaft Bearing
price	$133.39
firstName	Ket Wei
lastName	Yong
initial	K
address	3899 Retallack Street S4S 3B3
cardType	Visa
cardNum	• 123456789 • 123456789

Figure 3

⚠ Problem loading page   ✕ | 🌀 Applications   ✕ | Reading All Request Parameters ✕ | +

← ⓘ | localhost:8080/ReadAllParam/ReadParamServlet   ▽ | ⟳ | 🔍 Search

Reading All Request Parameters

Parameter Name	Parameter Value(s)
itemNum	123b
description	Drive Shaft Bearing
price	$133.39
firstName	Ket Wei
lastName	Yong
initial	k
address	*No Value*
cardType	MasterCard
cardNum	• 123456789 • 123456789

Figure 4

Question 2 (20 marks)

This exercise is to show that each servlet can only access its own initialization parameters set in the deployment descriptor. However, they can both access the same context parameter set in the deployment descriptor. You are to write two servlets, FamousQuoteServlet and GreetingServlet to demonstrate this.

The FamousQuoteServlet should print out a famous quote obtained from a servlet initialization parameter set in the deployment descriptor. The famous quote is "Put your hand on a hot stove for a minute and it seems like an hour. Sit with a pretty girl for an hour, and it seems like a minute". Your FamousQuoteServlet should try to access the initialization parameter of the GreetingServlet and show that it has no access to it by printing "no greeting". Finally, the FamousQuoteServlet should print the message set in the context parameter "I can access the context parameter". Figure 1 shows the output of the FamousQuoteServlet.

Do the same with the GreetingServlet. The output of GreetingServlet should look like Figure 2.

Both FamousQuoteServlet and GreetingServlet are in the same web application. Both servlets should be accessible with the URL as follow:
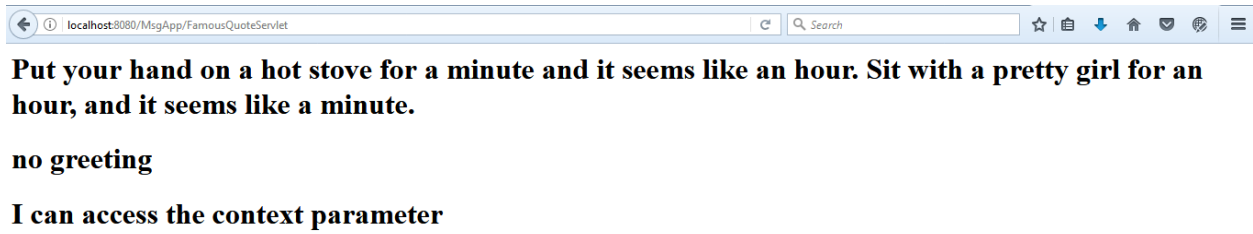
http://localhost:8080/MsgApp/FamousQuoteServlet
http://localhost:8080/MsgApp/GreetingServlet

Put your hand on a hot stove for a minute and it seems like an hour. Sit with a pretty girl for an hour, and it seems like a minute.

no greeting

I can access the context parameter

Figure 1

Happy Birthday

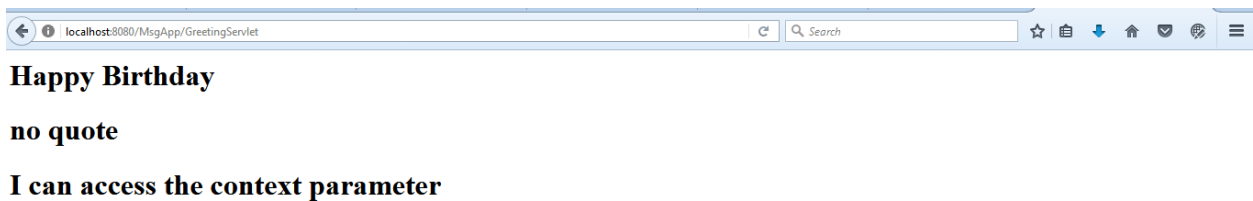**no quote**

I can access the context parameter

Figure 2

Question 3 (20 marks)

(A) Including static resource

This exercise is to show you that you can include static content from an external file in a servlet each time the servlet handles a request. You are to write a servlet called IncludeServlet to include an html page as shown in Figure 1 below. Your IncludeServlet should print out the message, Welcome To IncludeServlet and Imagine the rest of the page here, before including the CopyRight.html as shown in Figure 2 below.
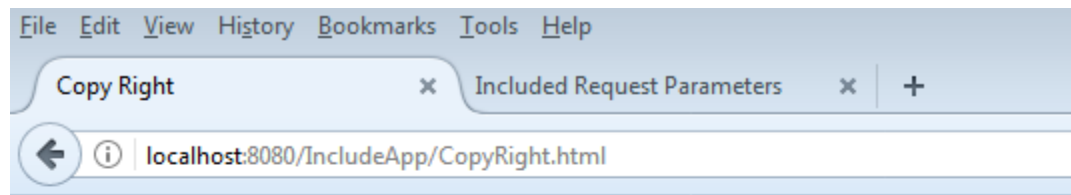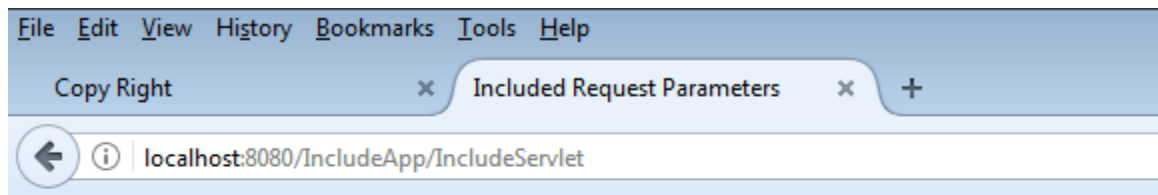
Figure 1

Figure 2

(B) Including dynamic resource

This exercise is to show you that you can include dynamic content generated by another servlet in the application. You are to write a servlet called FirstServlet to include the contents generated by a servlet called SecondServlet. Your FirstServlet should print the message **Included Request Attributes** before including the SecondServlet as shown in Figure 3 below. The SecondServlet should print out all the request attributes using the getAttributeNames() method of the HttpServletRequest. Your FirstServlet should set an attribute named Favorite Singer with value Taylor Swift by using a query string attaching to the SecondServlet's servlet path before dispatching the request to SecondServlet.
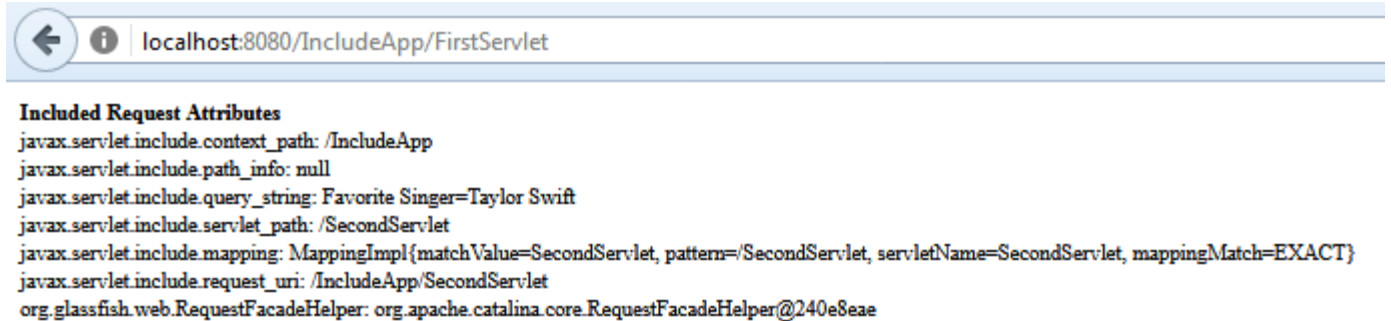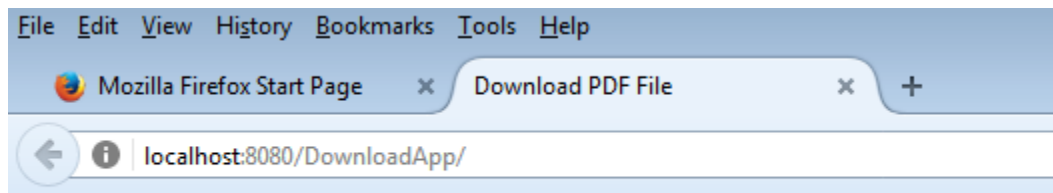
Included Request Attributes
javax.servlet.include.context_path: /IncludeApp
javax.servlet.include.path_info: null
javax.servlet.include.query_string: Favorite Singer=Taylor Swift
javax.servlet.include.servlet_path: /SecondServlet
javax.servlet.include.mapping: MappingImpl{matchValue=SecondServlet, pattern=/SecondServlet, servletName=SecondServlet, mappingMatch=EXACT}
javax.servlet.include.request_uri: /IncludeApp/SecondServlet
org.glassfish.web.RequestFacadeHelper: org.apache.catalina.core.RequestFacadeHelper@240e8eae

Figure 3

Question 4 (20 Marks)

Write an HTML form to let user enter a PDF file name, as shown in Figure 1 below. Write a servlet called DownloadServlet that can send a PDF file to the browser given a filename with or without the file extension PDF. After user enters the file name, clicking the Download button will invoke the DownloadServlet to send the PDF file to the browser or to download the PDF file.

The DownloadServlet should cater for the following scenario:
(i) If the user clicks the Download button without entering a file name or entering a blank file name, the DownloadServlet should resend the page in Figure 1 with a message "Please enter a valid file name" with the page titled "Blank Filename" as shown in Figure 2 below.
(ii) If the user enters an invalid file name such as: abcde, the DownloadServlet should resend the page in Figure 1 with a message "Filename: abcde.pdf not valid" with the page titled "Invalid Filename" as shown in Figure 3 below.

You should assume there is only one directory where all the PDF files are stored. This directory path should be set in the deployment descriptor (web.xml) file. The DownloadServlet should retrieve this path before being able to access the file.
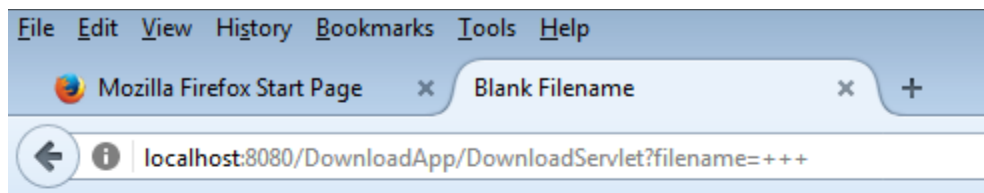
Figure 1

Figure 2

Figure 3

Question 5 (20 marks)

Extend the banking application posted in the UR Courses we discussed in the class to include a change password page to let user to change password. Your banking application should continue to assume that there will always be only three users, ann, john and mark exist in the application. Your application main page when started up should be as shown in Figure 1. Notice that you must add a link to let user change password. When user clicks the change password link, a change password page as shown in Figure 2 should appear. After user clicks submit to change the password, the main page should appear again to let user login. However, the main page should print a message to inform user whether the password change was successful or not, as shown in Figure 3 and 4. Upon successful login, a page displaying account status and a link back to main page should appear as shown in Figure 5. Your new banking application should allow user to change password as many times as he wish. The only dead end for this application is if users forget their password in the login page.
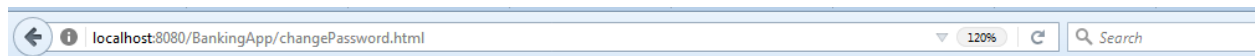
The Servlet Model

We talk about the basic concepts of the Servlet model. This web application contains the following examples from the slide.

1. This web application also implements the simple banking application explained in the slide. Here is the Login page for the application
   As explained, the usernames/passwords are hard coded in the LoginServlet. You can use any of the following values:
   1. ann/aaa
   2. john/jjj
   3. mark/mmm

   Once you login, you should see the account information of the user.
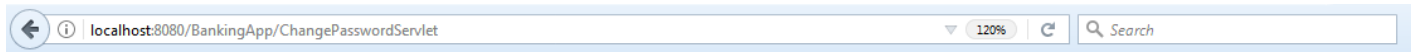
Change Password

Figure 1

Change Password

Username [   ]

Old Password [   ]

New Password [   ]

Confirm New Password [   ]

[Submit] [Reset]

Figure 2

Password Changed Successful
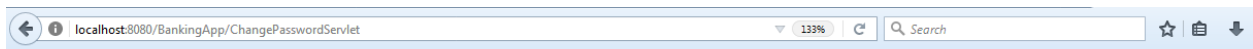
The Servlet Model

We talk about the basic concepts of the Servlet model. This web application contains the following examples from the slide.

1. This web application also implements the simple banking application explained in the slide. Here is the Login page for the application
   As explained, the usernames/passwords are hard coded in the LoginServlet. You can use any of the following values:
   1. ann/aaa
   2. john/jjj
   3. mark/mmm

   Once you login, you should see the account information of the user.

Change Password

Figure 3

Password Changed Unsuccessful

The Servlet Model

We talk about the basic concepts of the Servlet model. This web application contains the following examples from the slide.

1. This web application also implements the simple banking application explained in the slide. Here is the Login page for the application
   As explained, the usernames/passwords are hard coded in the LoginServlet. You can use any of the following values:
   1. ann/aaa
   2. john/jjj
   3. mark/mmm

   Once you login, you should see the account information of the user.
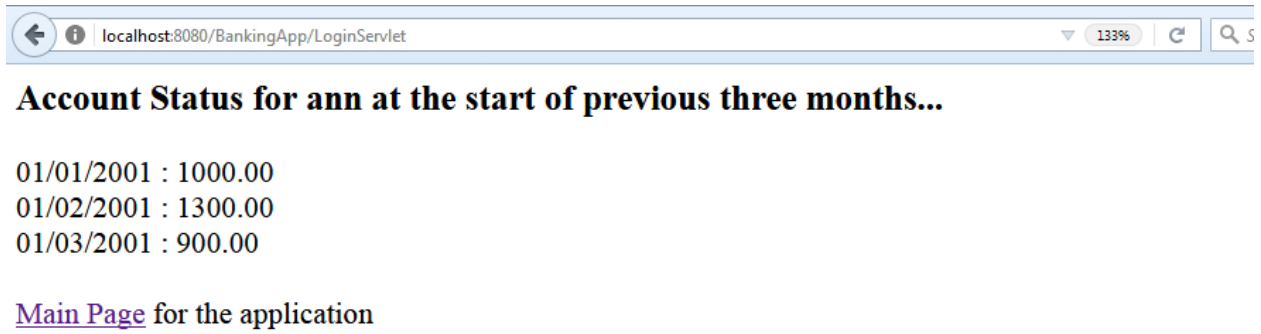
Change Password

Figure 4

Account Status for ann at the start of previous three months...

01/01/2001 : 1000.00
01/02/2001 : 1300.00
01/03/2001 : 900.00

Main Page for the application

Figure 5