CGS4854 PROGRAM 4 CHAPTER 4

Website Construction and Management
Professor: Michael Robinson
e-mail : michael.robinson@fiu.edu
Web Page : www.cs.fiu.edu/~mrobi002/teaching

- Your program must be named: yourLastNameFirstLetterOfYourFirstNamep1
  If your name is George Washington your tutorial should be named:
  WashingtonGp1

- Turn in the signed source code on paper, and email me the source code.
- Make sure the program is properly documented and aligned uniformally, looking professionally,
  I will take points off if it not.
- Include the following header in every program:

```
<%-- *****************************************************************************
Author     : Your Name
Professor  : Michael Robinson
Program #  : Assignment Purpose/Description
             {A brief description of the program }

Due Date   : MM/DD/YYYY
Certification:
I hereby certify that this work is my own and none of it is the work of any other person.

 ..........{ your signature }..........

****************************************************************************** %>
```

If you had any errors in Homework 3, then fix them, so that you don't lose points twice for the same mistake.

Enhanced Controller
===================

Create a new package for this assignment in your username web app.
It should start with the same fields, pages and validations that were required for Homework 3.

- Do not change the files that were used for assignment 3.
  - Assignment 3 should still be accessible from the web app.
  - You may copy files from assignment 3 into this new package.
  - You may import files from assignment 3, if they are unchanged.

- Modify the application so that it implements all the features from Chapter 4. Review the steps in Tutorial 4.
  - Add all the necessary files to the shared package.
  - Note that HellperBaseCh4 should be placed in the shared package.
  - Add all the necessary JAR files to the Libraries folder.

- Set up the log file properly.
  - Create the error.log file in the WEB-INF -> logs folder.
  - Edit web.xml and add the servlet definition for InitLog4j. Be sure it is called whenever the web app is loaded.

- Bean
  - Place the bean in the same package as the controller.
  - Start with the bean that was used in Homework 3.
  - Think of a calculation that needs to be done for your site. The calculcation should include a double and an integer. Think of a calculation that fits the theme of your site.
    Some examples might be:
      cost = price * quantity
      average price = cost / quantity
      total grade points = grade points * credits
      gpa = total grade points / credits
      total storage = megabytes * disks
      average storage = total megabytes / disks
      discount = total price * discout rate
      shipping cost = total pounds * rate

handling fee = items * fee

- Add a bean property that is a double.
  - This corresponds to the double that is used in your calculation.
  - Validate that the number is between reasonable minimum and maximum values. BeanUtils will set the value to 0 if a non-number is entered by the user. Be sure that 0 is not a valid value in your validation.
  - Provide a valid default value if the data is invalid.
- Add a bean property that is an int.
  - This corresponds to the integer that is used in your calculation.
  - Validate that the number is between reasonable minimum and maximum values. BeanUtils will set the value to 0 if a non-number is entered by the user. Be sure that 0 is not a valid value in your validation.
  - Provide a valid default value if the data is invalid.
- Add an accessor that returns a double.
  - There will be no mutator for this property (do not add a setter).
  - There will be no variable for this field. Do not add a varible for this field in your bean.
  - The accessor will return the result of the calculation.
  - This type of accessor is known as a calculated field.
- Add a logger to the Bean
  - Set the level of the logger to info.
  - Write an info message every time the setter of one of your fields is called.

- Controller Helper
  - Use a unique name, other than "helper", to save the controller helper in the session.
  - Change the name of the getData method to another name (it must start with get).
  - Write a warning message to the logger every time that the edit method is called. Include the value of the editButton from the query string in the message.
  - Write an info message to the logger every time that the copyFromSession method is called.
  - Write a debug message to the logger every time the process page is called. Include the value of one of your properties in the debug message.

- JSPs
  - There should be no hidden fields in any of the JSPs.
  - The JSPs should be in a hidden directory, but not in the same directory as the controller. Move the JSPs to an appropriate subdirectory of WEB-INF (not under classes).
  - Modify all the EL statements so that they refer to the new name that you used to store the helper in the session.
  - Modify all the EL statements so that they do not use data, but use the appropriate name for the new accessor method.
  - Add a new field to the edit page that will hold a double. This field will correspond to the new bean property that is a double.
  - Add a new field to the edit page that will hold an int. This field will correspond to the new bean property that is an int.
  - Echo these new fields in the confirm, wish list and process pages.
  - Echo the result of the calculation in the process page only. Use the bean property that only has an accessor.

- Run the application with different levels for the logger. Change the logger level in the constructor of the controller helper (do not modify the HelperBase).
  - Run the application with level set to debug, then warn, then info, then error. Review the log file after each run.
View the log file in a separate editor, as NetBeans might not re-read the file from the disk.
  - Be sure there is a hypertext link in the index.jsp page to the controller. Use a relative link.
  - N/A The web app must use the secure web.xml file and should only allow you and me to access your application: Edit tomcat users
  - Be sure to modify the project so that the .java files are placed in the WAR file.
  - N/A After the app is running in NetBeans, upload it and run it on ocelot: Uploading a WAR file