PROG2240: Java Web Tech
Assignment 2
Winter 2017

Assignment Type: Pair Programming

Due Date: Week 9 (start of class)
Note: Due on the first scheduled class after the Student Success Week

---

Before you begin:

- Do the Individual Exercises (A2_Exercise.pdf) and demo to your partner.
- Print **LastPage_PPLog.pdf**. Date and sign Sections B and C of your **Pair Programming (PP) Experience Log.**
  - Read the *Important Notes* section of A1.
- Continue to use your XXYYClub NetBeans project from A1.
  - Add **JSTL 1.2.x** library to the project
  - Do not modify the original main.css.

Background:

In this assignment, you will use JSP, EL, JSTL and Servlets using the **MVC pattern** to create a series of pages that allow you to view list of books, and add a book for the XXYY Programming Club. You will use ***server-side data validation*** and use the **web.xml** file to implement custom error handling.

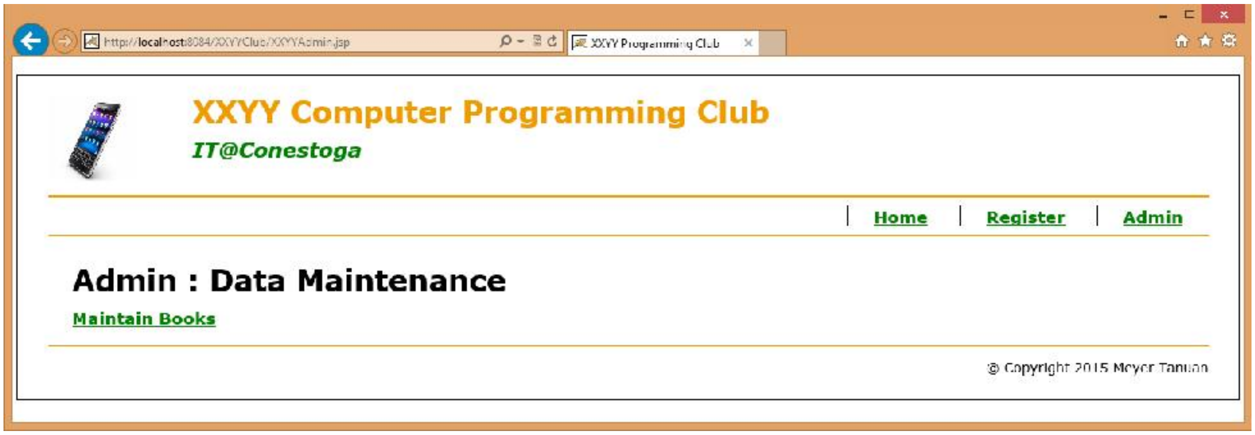
Additional Requirements:

- Use JSP exclusively as the “view” of the MVC pattern. JSP EL must use **attributes** (i.e., server variable).
  - Do not use input parameters (i.e., do not use ${param} EL).
- If possible, do not use JavaScript for page navigation and data validation.

Tasks

1. Add a new Admin JSP page

- Add a JSP page named XXYYAdmin.jsp
- Add the banner and footer pages.
- Add a new hyperlink: Link Maintain Books to XXYYDisplayBooks (URL pattern for servlet)



XXYY Computer Programming Club

IT@Conestoga

| Home | Register | Admin

Admin : Data Maintenance

Maintain Books

© Copyright 2015 Meyer Tanuan

---

2. Add Java classes to XXYYClub NetBeans project

- Under the Java source (i.e., "Source Packages") directory, add three Java packages: **club.admin, club.business, club.data.**
- Download **A2Files.zip** and copy these two Java classes to the *appropriate* packages.
  - club.business.Book.java
  - club.data.BookIO.java
  - ***Important:*** Do not modify any of these Java classes.
- Study all the code provided.
- Use the methods provided as part of your solution (see next steps).

---

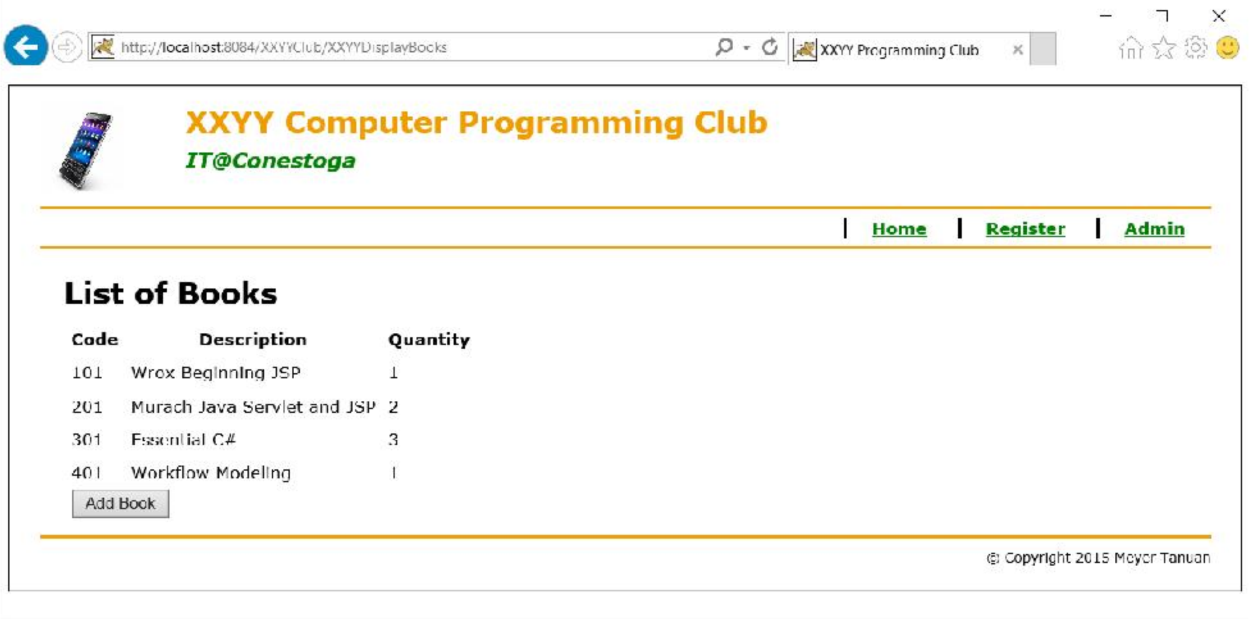
3. Add a servlet to read from books.txt and display the list of books

- From your downloaded **A2Files.zip**, copy **books.txt** to the **Web Pages/WEB-INF** folder
- Under **club.admin** package, add a servlet named XXYYDisplayBooksServlet.java

  - Set URL pattern to /XXYYDisplayBooks
- Within the **processRequest()** generated method:
  - Read all the books and store in the **request** object
    - *Hint:* Call getBooks() method of BookIO class
  - Forward the control to XXYYDisplayBooks.jsp page.

4. Add a JSP page to display books

- JSP page name: XXYYDisplayBooks.jsp
- Add the banner and footer pages.
- Add the code to match the design layout as shown below.
- Display all the books in a table.
  - *Hint*: Use the ArrayList of Book objects stored in the **request** object. Also see individual exercises (A2_Exercise.pdf).
- Add the "**Add Book**" button which calls the XXYYAddBook.jsp (Step 5)


http://localhost:8084/XXYYClub/XXYYDisplayBooks
XXYY Programming Club
| Home | Register | Admin
Add Book

XXYY Computer Programming Club

IT@Conestoga

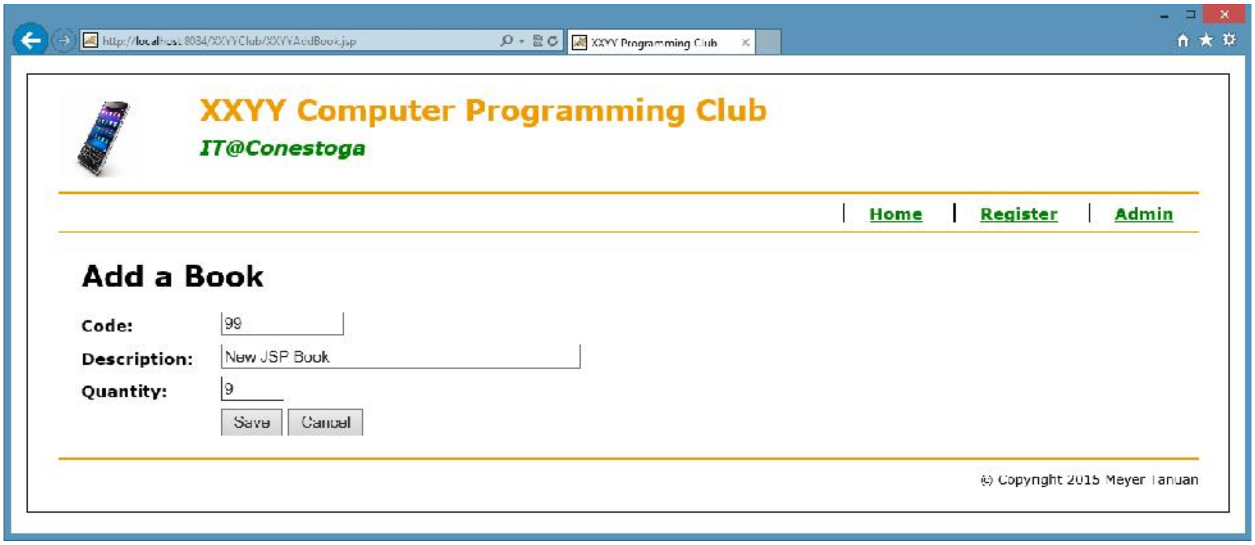
List of Books

Code	Description	Quantity
101	Wrox Beginning JSP	1
201	Murach Java Servlet and JSP	2
301	Essential C#	3
401	Workflow Modeling	1

© Copyright 2015 Meyer Tanuan

5. Add a JSP page to add book

- JSP page name: XXYYAddBook.jsp
- Add the banner and footer pages.
- Add the code to match the design layout shown below.
- All input fields must be of type "text"
- Add the "**Save**" button which calls the XXYYAddBook servlet with POST method type.
- Add the "**Cancel**" button which calls XXYYDisplayBooks servlet and will display a fresh copy of a list of books from the books.txt file.
  - *Hint*: You can add HTML5 attributes *formaction* and *formmethod*. See: www.w3schools.com/tags/tag_input.asp



XXYY Computer Programming Club

IT@Conestoga

Add a Book

Code: 99

Description: New JSP Book

Quantity: 9

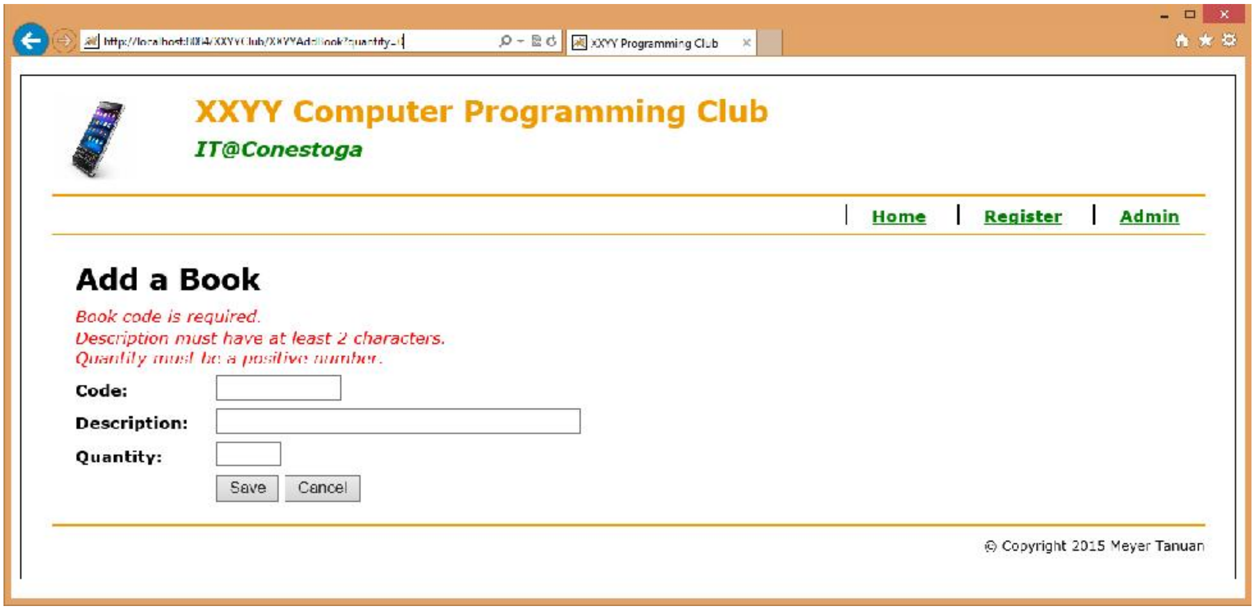
(c) Copyright 2015 Meyer Tanuan

6. Add a servlet to add a book (with server-side input validation)

- Under **club.admin** package, add a servlet named XXYYAddBookServlet.java
  - Set URL pattern to /XXYYAddBook
- Get the input parameters and create a Book object
- **Server-side input validation:**
  - If there are multiple errors, show them all at once (See sample screen shot below).
  - If **code** is blank, show the error message (in red font):

        - *Book code is required.*
    - If **description** does not have a minimum length of 2, show the error message:
        - *Description must have at least 2 characters.*
    - If **quantity** is negative or zero, show the error message:
        - *Quantity must be a positive number.*
        - Note: If the quantity entered is non-numeric, show the custom JSP error page (Step 7)
- If there are errors, use the **request** implicit object to store the Book object and the error messages. Then forward to the XXYYAddBook.jsp page.
    - *Note*: Your JSP page must "remember" the previous data entered by the user.
- **If the input fields are valid:**
    - Add the book to the text file.
        - *Hint*: Call the **insert()** method of **BookIO** class.
    - Forward to the XXYYDisplayBooks servlet.
        - Note: This will display the newly added book on the list.

    - **Test your code with this URL:** http://localhost:8084/XXYYClub/XXYYAddBook?quantity=0



Add a Book

Book code is required.
Description must have at least 2 characters.
Quantity must be a positive number.

Code:
Description:
Quantity:

7. Add a Custom JSP error page

a. Modify the web.xml to forward exceptions to a JSP error page

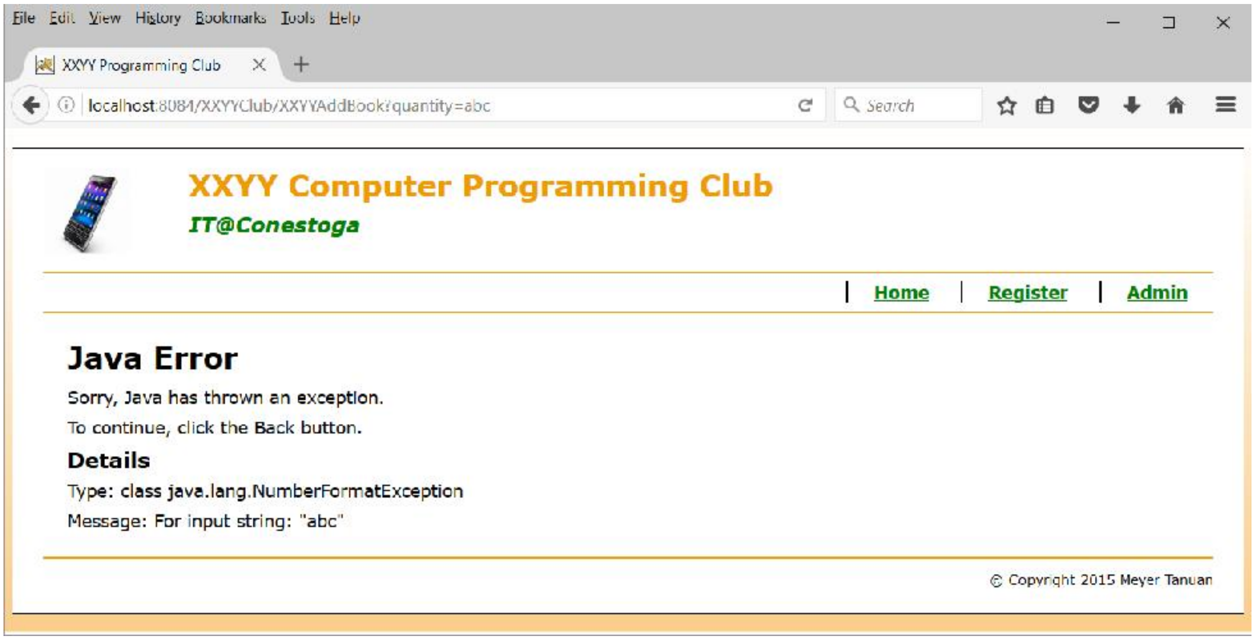
```
<error-page>
<exception-type>java.lang.Throwable</exception-type>
<location>/XXYYError.jsp</location>
</error-page>
```

b. Add an error JSP page

- JSP page name: XXYYError.jsp
- *Hint*: Copy part of error_java.jsp page from Exercise 5-1.

c. Test the error JSP page

- Try adding a book with non-numeric quantity
- Test your error page by executing a negative test case
  - Hint: http://localhost:8084/XXYYClub/XXYYAddBook?quantity=abc


File Edit View History Bookmarks Tools Help
XXYY Programming Club
localhost:8084/XXYYClub/XXYYAddBook?quantity=abc
Search
Home
Register
Admin

XXYY Computer Programming Club

IT@Conestoga

Java Error

Sorry, Java has thrown an exception.

To continue, click the Back button.

Details

Type: class java.lang.NumberFormatException

Message: For input string: "abc"

© Copyright 2015 Meyer Tanuan

8. Demo your Web Application

- Print and submit sample screen shots (similar to the five screen shots above) of your running application.
- Be ready to demo all the features of your assignment solution. Use the above steps only as a starting point.
- **Tips:**
  - Ensure values are "remembered" if fields are not valid.
  - Ensure your application does not crash when the books.txt is not found, etc.

Assignment Submissions

Important: You must submit your printouts (Cover Page, screenshots, PP Log) and upload your solution (XXYYClub.zip). Do not print your code.

Sequence	Item
1	A printout of the A2_CoverPage.pdf (a total of 3 pages) with all the relevant sections filled in.
2	A printout of 5 sample screenshots similar to the above screenshots (with the **title** displayed). The screenshots should be large enough for marking (Use Alt-PrtSc to copy and paste screenshot to MS Word, one screenshot per page).
3	A printout of the LastPage_PPLog.pdf with all the relevant sections filled in.
4	Zip and upload your assignment solution XXYYClub.zip (where XXYY is your team initials in upper case letters) to your appropriate section's **A2_Dropbox** on **eConestoga**. Upload using the **driver's** eConestoga login. **Note:** This zip file must be ready to run on NetBeans 8.1 without modifications when unzipped to C:\murach in any Doon IT Lab. *Otherwise you will get 5% penalty per upload-related error.*
5	Demo your assignment solution **on or before** the due date. Both pair programming team members must be present to demo your assignment. The instructor will randomly decide who will demo your solution. *The instructor will also randomly ask any member to demo their individual exercises solution.*