Deadline
Due by 11:59 pm on Sunday, April 3, 2016

How to submit
Submit your files for this homework using ~st10/328submit on nrs-projects, with a hw number of 8.

Purpose
To practice with PHP and sessions and Oracle.

Important notes
• NOTE: you are welcome to use require_once and include_once in your PHP documents! BUT when you use them in homework problems' documents, be sure to also SUBMIT copies of all files that you are requiring/including!
• Also note: I hope to have you present versions of some of these documents to the class at some point.
• Now there is an introductory set of CS 328 PHP Coding Standards, as discussed in class and as given on the public course web site -- still quite subject to change, so keep an eye on it.
• Remember to follow the CS 328 SQL and PL/SQL Coding Standards as given in the CS 328 Homework 2 handout and the public course web site for all SQL and PL/SQL code.
• Make sure that you have executed the scripts create-bks.sql and pop-bks.sql.
• Remember to follow the CS 328 HTML5 Coding Standards as discussed in class and as given on the public course web site for all HTML5 documents.
• Remember to follow the CS 328 CSS3 Coding Standards as discussed in class and as given on the public course web site for all CSS3 that you write, also.

Problem 1
As a warmup, there is a complete version of the Week 9 Lab's dept-fun.php and associated files on the public course web site, under "In-class examples"

You started your version of dept-fun.php in lab -- now complete it, making sure that your name appears in the h1 heading that appears atop each page output by dept-fun.php.

Also make sure it includes in its opening comment the URL one can use to run your dept-fun.php on nrs-projects.

Submit your dept-fun.php and all files it requires.
Problem 2

First: consider your external CSS3 file bks.css from the end of Homework 7. Make a new copy of this in a different directory, since you might be modifying it for this Homework 8 and you don't want to accidentally change how it styles Homework 7's files using it (since that could affect your Homework 7 grade!)

Now consider Homework 7 - Problem 5's get-order-status.php.

For Homework 8, create a new version of this named get-order-status-8.php, styled using bks.css, which meets the following specifications:

- This document should be one of those that can generate all 3 of this little application's pages, depending on the keys that exist in the $_POST and $_SESSION arrays when it is reached.
- Each page that this document generates should include, in its body element, your name, CS 328, and the name of your bookstore from bks-splash.html.
- This page should use your Homework 8 copy/version of bks.css (and you may add to it if you wish, but make sure the resulting version works with all of Homework 8's bookstore-related PHP documents).
- Its initial form should require that the user enter JUST an Oracle username and password.
- The action for the login form, when submitted, is to use the form-provided information to build a form including a dynamically-populated drop-down box of order numbers for the user to choose from.
- The action for that order-number-drop-down form, when submitted, is to use the form-provided information and appropriate session attributes to query the database for:
  - the date the specified order was placed
  - the publisher for this order
  - the book title(s) in this order
  - the quantity received to-date for each book in this order
  - ...and, for this homework, this is considered the end of this logical session.
- Further requirements for the handling-a-submitted-order-number part:
  - USE external PHP helper functions for each page of your application, as demonstrated in try-trio.php and dept-fun.php! (and additional helper functions/require files are encouraged, also!)
  - Make sure that you use a bind variable appropriately for the desired order number -- and you can assume that none of the order numbers should legitimately contain HTML tags, and so you can strip any that someone attempts to include.
  - There COULD be more than one row in response to this (multiple titles involved in a single order) -- there also could be zero rows (for a non-existent order number).
    
    You are expected to use a table element for displaying at least the book title(s) and their quantity received to-date for each; I'll leave it up to you whether you'd like to include the order data and publisher within this table element, or separately.
    
    Either way, display these results in an attractive, readable form; it is OK if you end up giving a table header row and no table content rows in the no-rows-found case.

- The response page to the order-number-drop-down form should also include a hypertext link with
appropriate text that links back to get-order-status-8.php (which should result in the login form being displayed, since the logical session is considered to be ended).

Do NOT add a link to your get-order-status-8.php from your index.html on nrs-projects; everyone is practicing pretty much the same thing here. DO include the URL one can use to run your get-order-status-8.php on nrs-projects in its opening comment.

Submit your get-order-status-8.php and all files it requires.

Problem 3

Consider Homework 7 - Problem 6's hand-insert-order-needed.php.

For Homework 8, create a new version of this named hand-insert-order-needed-8.php, styled using bks.css, which meets the following specifications:

• This document should be one of those that can generate all 3 of this little application's pages, depending on the keys that exist in the $POST and $SESSION arrays when it is reached.

• Each page that this document generates should include, in its body element, your name, CS 328, and the name of your bookstore from bks-splash.html.

• This page should use your Homework 8 copy/version of bks.css (and you may add to it if you wish, but make sure the resulting version works with all of Homework 8's bookstore-related PHP documents).

• Its initial form should require that the user enter JUST an Oracle username and password.

• The action for the login form, when submitted, is to use the form-provided information to build a form including a dynamically-populated drop-down box of ISBN's-and-titles for the user to choose from, BUT note:
  – the drop-down should DISPLAY to the user, for each option, an ISBN and its title,
  – BUT! the value submitted for each option should be JUST the ISBN for that option.

...and this form should also include a way for the user to enter the desired order quantity for the suggested order-needed.

• The action for that ISBN-drop-down form, when submitted, is to use the form-provided information and appropriate session attributes to update the order_needed table as specified;

...and, for this homework, this is considered to be the end of this logical session.

• Further requirements for the handling-a-submitted-ISBN-and-desired-order-quantity part:
  – USE external PHP helper functions for each page of your application, as demonstrated in try-trio.php and dept-fun.php! (and additional helper functions/require files are encouraged, also!) (and of course submit all of those files also!)
  – You are required to call Homework 4, Problem 1's PL/SQL stored procedure insert_order_needed (an example version of which is available on the course Moodle site under "Selected solutions" if you have any qualms about your version)
  – Make sure that you use bind variables appropriately for the desired ISBN and desired quantity in your stored procedure call --

and since neither of these should legitimately contain HTML tags, you can reasonably strip any that someone attempts to include.
- Make sure you commit the result!
- Also display a descriptive message describing the order-needed insertion in a readable/reasonably attractive way.

**The response page to the isbn-and-desired-quantity form should also include a hypertext link with appropriate text that links back to** hand-insert-order-needed-8.php (which should result in the login form being displayed, since the logical session is considered to be ended).

**Do NOT** add a link to your hand-insert-order-needed-8.php from your index.html on nrs-projects; everyone is practicing pretty much the same thing here. **DO** include the URL one can use to run your hand-insert-order-needed-8.php on nrs-projects in its opening comment.

Submit your hand-insert-order-needed-8.php and Homework 8's bks.css and all files they require.

**Problem 4**

Consider the "additional database" you committed to in Homework 7, Problem 1. Decide on a useful action for someone in its scenario that would involve dynamically querying and creating a form control based on that database, and then letting the user do something useful based on what they choose from that form.

**First:** consider your external CSS3 file custom.css from Homework 7. Make a **new** copy of this in a **different** directory, since you **might** be modifying it for this Homework 8 and you don't want to accidentally change how it styles Homework 7's problem using it (since that could affect your Homework 7 grade!)

For Homework 8, create PHP document custom-session1.php, styled using custom.css, which meets the following specifications:

- This document should be one of those that can generate all 3 of this little application's pages, depending on the keys that exist in the $_POST and $_SESSION arrays when it is reached.

- Each page that this document generates should include, in its body element, your name, CS 328, and the name of your additional database's scenario.

- Its initial form should require that the user enter JUST an Oracle username and password.

- The action for the login form, when submitted, is to use the form-provided information to dynamically-populate-and-build a form control within a resulting form for the user to respond to (along with other appropriate form controls for your situation).

- The action for that form-with-a-dynamically-populated-part, when submitted, is to use the form-provided information and appropriate session attributes to do some action useful to the user, ...and, for this homework, this is considered to be the end of this logical session.

- The response page to the form-with-a-dynamically-populated-part should also include a hypertext link with appropriate text that links back to custom-session1.php (which should result in the login form being displayed, since the logical session is considered to be ended).

- **USE** external PHP helper functions for each page of your application, as demonstrated in try-trio.php and dept-fun.php! (and additional helper functions/require files are encouraged, also!) (and of course submit all of those files also!)

**DO** add a link to your custom-session1.php from your index.html on nrs-projects, and **ALSO** include the URL one can use to run your custom-session1.php on nrs-projects in its opening
Submit your `custom-session1.php` and Homework 8's `custom.css` and all files they require.