

CS 328 - Exam 2 Review Suggestions - Spring 2016

last modified: 2016-04-01

- You are responsible for material covered in class sessions and homeworks; but, here's a quick overview of especially important material.
- You are permitted to bring into the exam a single piece of paper (8.5" by 11") on which you have **handwritten** whatever you wish on one or both sides. This paper must include your name, it must be handwritten by you, and it will **not** be returned.
 - Other than this piece of paper, the exam is closed-note, closed-book, and closed-computer.
- You are still responsible for SQL! (and PL/SQL, and HTML5, and CSS3, and other Exam 1 topics).
 - (obviously, we are building on and making use of previous material in much of this new material.)
 - there will be at least one question focused **purely** on SQL.
 - I hope there will also be some questions involving how the new material fits in the n-tiered architecture discussed earlier in the semester; that's a theme we are discussing throughout the semester.
 - but, in general, the FOCUS of most of the questions on this exam will be the material covered since Exam 1.
- This will be a pencil-and-paper exam, but you will be reading and writing code, statements, and expressions in this format. There will be questions about concepts as well.
- A packet of example code will be given out along with the exam, both for reference and for use directly in some exam questions. Because of the nature of this code (some being used directly in exam questions, for example), it cannot be made available in advance -- however, it will happen to include the following:
 - an uncommented version of `html5-template.html`
 - examples of a PL/SQL stored procedure, a PL/SQL stored function, and PL/SQL exception handling
 - example HTML5 that happens to include a hyperlink, a form, a textfield, a radio button, a checkbox, a drop-down box, and a submit button
 - examples of an external CSS3 file and an external PHP file
 - an example HTML5 document using external CSS3, and using external PHP
 - an example of connecting to Oracle from PHP
 - an example of executing a SQL query, PL/SQL stored procedure, and PL/SQL stored function from PHP
- Note that the ability to read and make use of existing code is an important skill.
 - It is possible that you may have to diagnose what is wrong with provided buggy code, and how

it might be fixed, and/or perhaps you could be asked to modify code.

- You might be asked to complete incomplete code (you could be given partial code, and asked to complete or modify or debug it in some way).
- Your studying should include careful study of posted examples and notes as well as the homeworks (and posted homework example solutions) thus far.
- You are responsible for material covered through and including Homework 8 and part of the security discussion from Week 10 Lecture 1.

CSS3, continued

- You are now more comfortable with more kinds of selectors and properties.
 - You should know how to write and use an id attribute selector, and why you might want to do so.
 - You should know how to write and use a context selector, and why you might want to do so.
- What is the CSS Box Model? What does it describe? You should be familiar with this, and with the "pieces" within it.
- Know the basics of using the float and clear properties to get multiple columns and sidebars. Know the basics for how to format forms.
- You are responsible for those CSS3 features that have been discussed in lecture and in lab, as well as those CSS3 features that have been used in posted course examples and in course assignments.

PHP

- What does PHP stand for? What is it?
- Consider an n-tier architecture. On which "tier" is PHP executed? Be comfortable with how a PHP-enabled document is handled/processed.
- What languages' syntax influenced PHP syntax?
- How would you name a PHP-enabled file? Where would you normally place a PHP-enabled file in Humboldt's set-up? What permissions does this file need to have there? What URL would you (or an HTML page) then use to access that PHP-enabled file?
- What is the preferred PHP tag for this course? That, along with PHP's expression tag, are the only PHP tags you are responsible for on the final (and the only ones you should use on the final... 8-))
- Should be comfortable with the PHP syntax and features discussed in class and used in exercises and assignments (including, but not limited to:
 - how do you write scalar variables? numeric literals? string literals?
 - how can you output a value?
 - how can you write a comment?
 - how can you concatenate strings? do basic arithmetic?

- how can you write a function? call a function?
 - how can you do branching, repetition?
- Keeping in mind that there are numerous means that one can use with PHP to allow it to interact with databases, which is the one we used to connect PHP to Oracle?
- What is the difference between the PHP functions `isset` and `array_key_exists`?
 - Which would be useful to see if a key exists in a PHP associative array?
 - Why, in the context of web applications, might it be useful to know if a key exists in a PHP associative array such as `$_POST` or `$_GET` or `$_SESSION`?
- make sure that you are comfortable with:
 - obtaining parameter values from form inputs
 - setting and obtaining session variables
 - connecting to an Oracle database, executing SQL statements, stored procedures, and stored functions, and retrieving results (as appropriate)
 - make sure you are also comfortable with appropriately using bind variables in SQL statements and stored procedure/stored function calls
- What is a server-side include (SSI)? What four PHP functions can you use to get this? Should be able to read and use these functions, and know the difference between them (and when one might be preferred to the others).

Finite state machines (FSMs)

- (also known as finite automata, deterministic finite state machines, deterministic finite automata)
- could give you a "classic" FSM, and ask you if given strings are accepted by that FSM; could ask you to name its start state, its accepting states; could ask you, given the current state and an input, what the resulting state/next state would be;
- you should be able to read and answer questions about a FSM representing a simple web application as well (in the style discussed in class)
- I will **not** ask you to draw a FSM for Exam 2 (maybe on the Final...)
- (you are only responsible for the "classic" FSM notation discussed in lecture; you are not responsible for UML-style FSMs.)

PHP and sessions

- what is the nature of HTTP with regard to state? Given **just** HTTP (that is, I'm not talking about PHP or cookies or other features external to HTTP), can you associate a request with a previous request?
- Should be comfortable with sessions in PHP. To use session attributes, what do you have to do in a PHP document (and when)? How can you set session attributes? How can you retrieve session attributes? How can you invalidate a session?

- Should be comfortable with single PHP files that handle a multi-page session (such as `dept-fun.php` and `try-trio.php`, and as you have practiced in homework problems)

XSS and SQL Injection

- We discussed two important vulnerabilities to defend against in web applications:
 - XSS - cross-site scripting
 - SQL injection
- What is XSS (cross-site scripting)? When is an application vulnerable to this?
- What is SQL injection? When is an application vulnerable to this?
- **Know** that both of these can occur when the application-server-tier programmer does not appropriately validate input fields.
- According to the Open Web Application Security Project (OWASP), what is the best attitude for a web-server-side programmer to take, with regard to untrusted data?
- Why can't client-side validation of data suffice in protecting against such attacks?
- What, then, are some of the approaches for preventing XSS/cross-site scripting?
- What, then, are some of the approaches for preventing SQL injection?