CS 318 - Final Exam Review Suggestions - Spring 2013

last modified: 5-8-13

- Remember: YOU ARE RESPONSIBLE for material covered in class, lectures/labs, and especially anything that's been on a homework or lab exercise; BUT, here's a quick overview of especially important material.
- final is CUMULATIVE!
  - if it was fair game for Exam 1 or Exam 2, it is fair game for the final;
  - Thus, using the posted review suggestions for Exam 1 and Exam 2 in your studying for the final would be a good idea. (Note that they are still available from the public course web page, under "Homeworks and Handouts".)
  - studying your Exam 1 and Exam 2 would also be wise.
  - there may indeed be similar styles of questions on the final as on those exams.
- 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.
- Final exams are not returned, although they will be kept on file for at least 2 years, and you are welcome to come by my office to look over your graded exam once it has been graded.
- This will be a pencil-and-paper exam, but you will be reading and writing code, statements, and expressions in this format. There could also be questions about concepts, of course.
- Note that the ability to read and make use of existing code is an important skill.
  - Some code may be included along with the exam, both for reference and for use directly in some exam questions.
  - 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).
- LOOK at posted examples and notes; LOOK at your previous course assignments and exercises (from both lecture and lab); recall that there are posted example solutions for many of the homework problems.
- There will be at least one question focused purely on SQL.

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?

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 do you write a function? call a function?
  - how do 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?
- 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).
- 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 `try-trio.php`, and as you have practiced in homework problems)

A few words on Web Design, Usability, and Accessibility

- be familiar with the user interface design guidelines discussed in class; be familiar with the concepts from Chapter 7 in the course textbook.
- remember that the application screens should help the user tell the story of his/her task;
- what are some human factors that should be considered in interface design?
  - ...and some general characteristics of users that should be kept in mind?
- what are some rules of thumb for visual design of web applications? ...for organization? ...for page layout? ...for navigation and links? ...for forms and controls?
- what are some guidelines for making accessible web applications?
- common mistakes in interface design? some common web usability problems? some common content usability problems? some common link usability problems? some common feature usability problems?

A few words on XML and JSON

- what does XML stand for? What was XML designed to do?
- what is an important aspect of XML elements?
- What can XML be used for? Why might this be beneficial?
- you are expected to be comfortable with XML syntax; you should be able to recognize a well-formed XML document, you should be able to write a well-formed XML document.
  - what is meant by root element? child element?
  - what is meant by an element having simple content? element content? empty content? mixed content? Given an XML page, you should be able to identify which elements have each of these kinds of content; you should be able to write example XML elements containing each type of content.
  - what is meant by an attribute?
  - What is necessary for an XML document to be said to be a well-formed XML document?
  - What is necessary for an XML document to be said to be a valid XML document? (two things are required, note)
- What is the purpose of DTDs and XML Schemas?
- According to the W3C XML specification, what should a program do if it is processing an XML document

and it finds a syntax error? Why is this the case?

- What are some advantages of XML? What are some disadvantages?
- What does JSON stand for? What is JSON?
- What is JSON syntax based on/very similar to?
- What does `JSON.parse(myJSONData)` do? What does `JSON.stringify(myJavaScriptObject)` do?
- What is the the syntax for object literals in JavaScript? What are a couple of minor differences between JSON syntax and JavaScript object literal syntax?
- What are some advantages of JSON? What are some disadvantages?

Ajax

- what does Ajax stand for?
- which technologies are involved in Ajax? what is required of a browser for Ajax to work? is anything special required of a web server for Ajax to work?
- should be comfortable with the basic approach here, and how it differs from the standard web server-client browser behavior
- why might one choose to use Ajax? why might one choose to not use Ajax?
- what is asynchronous about Ajax?

Miscellany

- Should be able to discuss tradeoffs of choosing different approaches discussed this semester, and what kinds of considerations would arise with these different approaches. ("plain" HTML pages, dynamic pages generated by servlets, JSP's, PHP-enabled pages, using PL/SQL stored procedures and functions in conjunction with the preceding approaches, using PL/SQL stored procedures/stored functions versus SQL statements; and maybe more!)