CS 328 - Final Exam Review Suggestions - Spring 2025

last modified: 2025-05-07

Final Exam BONUS Opportunity

- You can receive (a maximum) ***5 POINTS BONUS*** on the Final Exam if you do the following:
  - Make a **hand-written** Final Exam study sheet (a single sheet of paper on which you have hand-written as much as you would like on one or both sides).
  - Submit a photo or scan of it saved as a `.pdf`, `.png`, `.jpg`, `.gif`, or `.tiff` to Canvas **by 3:00 pm on Monday, May 12** such that I can read at least some significant **CS 328 post-Exam-2-specific** material on it.
  - You are **encouraged** to have this **with you** as you are taking the Final Exam.
  - **NOTE**: if this is typed rather than handwritten, you will **not** receive bonus credit, and you will **not** be allowed to use it during the Final Exam.
  - Please let me know if you have any questions about this, and I hope it helps you in reviewing course concepts more effectively before the Final Exam.

Final Exam Set-up

- You will take the Final Exam in SH 108 at **3:00 pm** on **Monday, May 12**.
  - You are expected to work **individually** on the exam -- it is not acceptable during the exam to discuss anything on the exam with anyone else.
  - You may have your Final Exam study sheet and ***also*** your Exam 1 and Exam 2 handwritten one-page study sheets on hand during the exam. Otherwise, the exam is closed-note, closed-book, and closed-computer/closed-electronic-devices.
- The Final Exam is **cumulative**, covering all of this semester's course material.
  - If a topic was fair game for Exam 1 or Exam 2, it is fair game for the Final Exam.
  - So, studying the review suggestion handouts for Exam1 and Exam 2 would be wise; (they're still available from the course web page, under "Homeworks and Handouts").
  - This review handout is a quick overview of especially important material since Exam 2.
- I expect there will be a few multiple-choice questions, and the rest will be short- to medium-answer questions.
- Your studying should include careful study of posted examples, notes, and assigned zyBooks chapters thus far.
- You are responsible for material covered in class sessions, lab exercises, and homeworks.
  - TIP: It is **perfectly fine** to retake/read over the short-answer questions in Canvas from course Homeworks as you are studying for the Final Exam!

These are set up for unlimited retakes, and only keep the highest score, so you will not hurt your grade by doing so!

- 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.
- 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 at least the following:
  - an *uncommented* version of **`html-template.html`**
  - example of a PL/SQL stored function and of PL/SQL exception handling
  - example HTML that happens to include a hyperlink, a form, a label, a fieldset, a textfield, a radio button, a checkbox, a **`select`**/drop-down box, and a submit button
  - examples of an external CSS file, an external PHP file, and an external JavaScript file
  - an example HTML document using external CSS, external PHP, and external JavaScript
  - an example of executing a SQL query, PL/SQL stored procedure, and PL/SQL stored function from PHP using OCI
  - an example using JavaScript that happens to include setting an element's event handler when the window is loaded to make some change to the current document using the DOM when a particular event occurs
- While PL/SQL and SQL are not case-sensitive (except within quotes!), strict-style HTML, CSS, PHP, and JavaScript frequently are. You are expected to use the correct case (when applicable) in your answers.
- Note that you are also responsible for knowing -- and following -- the **course style standards and course coding standards** in all of your answers.
- 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).

PL/SQL triggers

- What is a PL/SQL trigger? For what might it be useful?
- You should be able to read and write PL/SQL triggers; you should be able to determine, and be able to describe, when they would be executed.
  - How do PL/SQL triggers differ from PL/SQL stored procedures and stored functions?
  - Can a PL/SQL trigger can have parameters?

  - What causes a PL/SQL trigger to be executed?
  - What is a type of SQL **`select`** statement that is not permitted within a trigger?
- What are **`:new.`**`col_name`, **`:old.`**`col_name`, and how can they be used in a trigger?
- Given a trigger, you SHOULD be able to read it and answer questions about it, including:
  - When it would be executed (note that there are **four** aspects of this you should be able to indicate: before or after, what action, on what table, and is this trigger executed for each row affected?)
  - What are the trigger's effects?

PHP and OCI bind variables

- Make sure you are comfortable with appropriately using OCI bind variables in SQL statements and PL/SQL stored procedure/stored function calls.
  - This includes writing them appropriately within the SQL or PL/SQL command string,
  - and also using **`oci_bind_by_name`** to bind a value to them before executing the command.
- Make sure you understand that just concatenating user-provided data into a SQL statement is dangerous, and prone to SQL Injection. (Also make sure you understand that including a regular PHP variable in a double-quoted string, using variable interpolation, is also considered to be concatenation with regard to being prone to SQL injection.)
  - Know how to use **bind variables** to more safely include user-provided data as part of a SQL statement.
  - (Note that a statement using these bind variables can also be more efficient, especially if that statement is executed more than once.)

PHP and using OCI to call stored procedures and stored functions

- You should know how to use OCI from PHP (on the application tier) to call a PL/SQL stored procedure (on the data tier).
- You should know how to use OCI from PHP (on the application tier) to call a PL/SQL stored function (on the data tier).
  - You should also know how to obtain the value returned by that stored function.

PHP and sessions

- What is the nature of HTTP/HTTPS with regard to state? Given **just** HTTP/HTTPS (that is, I'm not talking about PHP or cookies or other features external to HTTP or HTTPS), 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)?
- You should know how to add session attributes to the **`$_SESSION`** array, how to set the value for a session attribute within the **`$_SESSION`** array, and how to obtain the value for a session attribute from the **`$_SESSION`** array.
- How can you invalidate a session? Be sure that you explicitly do this as soon as a a session is logically completed.
- Be comfortable using the PHP function **`array_key_exists`** to see if a key exists in a PHP associative array.
  - Why, in the context of web applications, might it be useful to know if a key exists in a PHP associative array such as **`$_POST`** or **`$_SESSION`**?
- For Exam 2, you were expected to be comfortable with setting up a PHP postback document, one that can either create a form or respond to it, depending on how it is called.

  Now you should be comfortable with setting up a PHP postback document that can appropriately use **`$_SESSION`** to handle all of the stages of a three-or-more stage application.

Intro to (unobtrusive-style client-side) JavaScript

- What is the relationship between JavaScript and Java?
- JavaScript was initially designed to add interactivity to HTML pages; while it has now expanded to being able to do much more, we are focusing on so-called **unobtrusive-style client-side JavaScript** in CS 328.
  - When we mention "JavaScript" in this class, then, you should assume that unobtrusive-style client-side JavaScript is intended unless explicitly specified otherwise.
- Note that you are expected to use unobtrusive-style client-side JavaScript in your exam answers.
- Consider an n-tier architecture. On which "tier" is the style of JavaScript used in CS 328 executed (given the assumptions above)? Be comfortable with how a document containing JavaScript is handled/processed.
- What are some of JavaScript's capabilities?
- How would you name an HTML file containing JavaScript? Where would you normally place an HTML file containing JavaScript on nrs-projects? What permissions does this HTML file need to have there? What URL would you (or an HTML page) then use to access that HTML file containing JavaScript?
  - What if we are talking about an external JavaScript? How would you name that file? Assuming that you are using unobtrusive-style client-side JavaScript, how could you use an external JavaScript's contents within an HTML file? What would be the CS 328-required element for including it, and where should this element be placed (according to CS 328 course coding standards)?
- Should be comfortable with the JavaScript syntax and features discussed in class and used in exercises and assignments (including, but not limited to:

  - What is the CS 328-required way to write and use variables in JavaScript?
  - how do you write a comment within JavaScript?
  - how can you concatenate strings? do basic arithmetic?
  - how do you write a function? call a function?
  - how do you do branching, repetition?
  - what is the difference between == and ===?
- What is the meaning of the **`onload`** attribute of an HTML **`window`** element? ...of the **`onsubmit`** attribute of the HTML **`form`** element? ...of the **`onclick`** attribute of the HTML **`button`** element? How can these be used in conjunction with unobtrusive-style client-side JavaScript?
- Consider the DOM (Document Object Model) -- what is the **`document`** object?
  - How can you use its **`getElementById`** method to obtain a reference to a JavaScript object corresponding to a particular HTML element object within that page? What attribute should an HTML element have to allow it to work with this method?
  - How can you use such a corresponding JavaScript object to set the value of an *attribute* for an HTML element within that page?
  - How can you use such a corresponding JavaScript object to obtain or set the *content* of an HTML element within that page?
  - Understand how, in the head of an HTML page, you can have a JavaScript that can set the **`window`** object's **`onload`** attribute to an anonymous function that sets the event handlers for event-related attributes of HTML elements within that page.
- You should be able to use unobtrusive-style client-side JavaScript to set an element's event handler when the window is loaded.
- You should be able to use unobtrusive-style JavaScript to validate a form, and to prevent its submission if it is not filled out "appropriately".
- You should be able to use unobtrusive-style JavaScript to set an element's attribute, and to set an element's content (make sure you know the difference!).

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 data from the user (for example, from submitted forms).

- Make sure you know that PHP (and the application tier in general) **always** needs to **NOT** trust user input from the client tier, no matter how much your particular form may try to prevent user input errors by using client-side JavaScript and appropriate HTML form widgets.
  - 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?
  - Make sure you know that use of **bind variables** is a powerful tool in helping to fight SQL injection

Intro to XML and JSON

- There will be an example of a well-formed XML document and syntactically-correct JSON provided in the Final Exam reference packet.
  - Between that and the class discussion on Week 15 Lecture 1, you should be able to write a fragment of well-formed XML and syntactically-correct JSON for describing a small set of data.
- what does XML stand for? What was XML designed to do?
- What can XML be used for? Why might this be beneficial?
- you are expected to be comfortable with XML syntax; given a well-formed XML document, you should be able to write a fragment of well-formed XML.
  - 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 document, 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 does JSON stand for? What is JSON?
- What can JSON be used for? Why might this be beneficial?
- What is JSON syntax based on/very similar to?
- What does **`JSON.parse(myJSONData)`** do? What does **`JSON.stringify(myJavaScriptObject)`** do?
- What is the syntax for object literals in JavaScript? What are some differences between JSON syntax and JavaScript object literal syntax?

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.
  - what actions, validations, application logic, etc. can/should be done on the client tier? on one of the application tiers? on the data tier?

    ...which should be done on MULTIPLE tiers? (to improve usability or the user experience BUT also help guard against cross-site scripting or SQL injection, for example)
  - using "plain" HTML pages vs. using dynamic pages generated by PHP
  - using PL/SQL stored procedures and functions in conjunction with the preceding approaches
  - using PL/SQL stored procedures/stored functions vs. using "plain" SQL statements
  - (you've only used PHP on the application tier in CS 328, but you know there are other options, also -- Java servlets, Java JSP pages, Python, *server*-side JavaScript, and more! So, hopefully you will be in a reasonable position to experiment with and learn some of these other application tier options at your leisure.)