CS 328 - Exam 2 Review Suggestions - Spring 2025

last modified: 2025-04-07

Exam 2 BONUS Opportunity

- You can receive (a maximum) ***5 POINTS BONUS*** on Exam 2 if you do the following:
  - Make a **hand-written** Exam 2 study sheet (a single sheet of paper, no larger than 8.5" by 11", 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 Wednesday, April 16** such that I can read at least some significant **CS 328 Exam 2-specific** material on it.
  - You are **encouraged** to have this **with you** as you are taking Exam 2.
  - **NOTE**: if this is typed rather than handwritten, you will **not** receive bonus credit, and you will **not** be allowed to use it during Exam 2.
  - Please let me know if you have any questions about this, and I hope it helps you in reviewing course concepts more effectively before Exam 2.

Exam 2 Set-up

- You will take Exam 2 in SH 108 on Wednesday, April 16.
  - 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 Exam 2 study sheet with you during the exam. Otherwise, the exam is closed-note, closed-book, and closed-computer/closed-electronic-devices.
- 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 still responsible for SQL, PL/SQL, HTML, and other Exam 1 topics.
  - (We are building on and making use of previous material in much of this new material.)
  - 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.
- You are responsible for material covered in class sessions, lab exercises, and homeworks through and including Week 10 Lecture 2 (April 2), the Week 10 Lab Exercise (Thursday/Friday, April 3/4) and

Homework 9 (due 11:59 pm on Friday, April 11).

  - This review handout is intended to be a quick overview of especially important material since Exam 1.
  - TIP: It is **perfectly fine** to retake/read over the short-answer questions in Canvas from Homework 6 as you are studying for Exam 2!
    - These are set up for unlimited retakes, and only keep the highest score, so you will not hurt your grade by doing so!
  - This is likewise true for the zyBooks Chapters 3, 4, and 5 course text activities, although note that you will only receive credit for activities you have completed by 11:59 pm on Friday, April 11.

- While PL/SQL and SQL are not case-sensitive (except within quotes!), strict-style HTML, CSS, and PHP frequently *are*. You are expected to use the correct case (when applicable) in your answers.
- You are expected to follow CS 328 course style guidelines and coding standards in your answers.
- 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 HTML document that happens to include at least one of each of the following elements: anchor (hyperlink), form, submit button, label, fieldset, textfield, radio button, checkbox, `select`/drop-down box
  - Examples of an external CSS file and an external PHP file
  - An example HTML document using external CSS, and using external PHP
  - An example of connecting to Oracle from PHP
  - An example of executing a SQL query from PHP
- 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).

nrs-projects permissions, and URLs versus file names

- You should be getting more and more familiar with these, and you might be asked similar questions to those asked on Exam 1.
- Consider an HTML or CSS or PHP file on nrs-projects.
  - You should understand the difference between its absolute file name on nrs-projects and its

relative file name depending on your current working directory.

  - You should understand the difference between its absolute file name on nrs-projects and the URL (Uniform Resource Locator) you would use in a browser to access that file.

- Where would you normally place an HTML or CSS or PHP file on nrs-projects? What permissions does the HTML or CSS or PHP file need to have there? What permissions do all of the directories in that file's path need to have? What URL would you then use to access that HTML or PHP document or CSS file?
  - How could you write a `link` element within an HTML or PHP document to use an external CSS file that happens to be in the **same** directory as that document to style that document? ...that happens to be in a **different** directory from that document to style that document?
  - How could you write a PHP `require_once` or `require` or `include_once` or `include` within a PHP document to appropriately include the contents of another file in the **same** directory?
    - You should be able to choose which of these PHP functions is most appropriate for a given situation.

HTML note - `table` element

- You should be comfortable with using an HTML `table` element to structure tabular data (such as a `select` statement's results).
- You should be familiar with the course style guidelines for HTML `table` elements.

CSS - Cascading Style Sheets

- What does CSS stand for? What is its purpose?
- Consider an n-tier architecture. On which "tier" is CSS *executed*?
- What does "cascading" mean in CSS? what does "style" mean?
- What are some of the potential benefits of CSS?
- You should know **what** an external style sheet, an internal style sheet, and an in-line style sheet are, but you will **not** be asked to read or write *any* internal style sheets or in-line style sheets.
  - You **should** know **why** external style sheets are preferable to internal style sheets or in-line style sheets,
  - and you should expect to have to read external style sheets, and be able to write rules as they would appear within an external style sheet.
- How can one set up an external style sheet? What do you do to have an HTML page use an external style sheet? (Be sure to know the CS 328-approved HTML element for this.)
  - What do you do (if anything) to HTML elements within an HTML document to have external CSS rules apply to them? (This varies based on the rule selector, of course.)
- What is the purpose of `normalize.css`?

- What is the basic CSS rule syntax?
  - What are the main parts of a CSS rule? (a selector followed by a declaration block)
  - What is a property? What is a property value? How are these written within a declaration in a rule's declaration block?
  - Given a set of CSS rules, you should be able to predict how a given HTML document or given HTML elements would be displayed;
  - Should be able to read, write such rules;
  - Should be able to write an HTML element such that it would be styled by a CSS rule;
  - What does a selector do/indicate? what does a property do/indicate? what does the value (in a style) do/indicate?
- If rules within an external style sheet conflict, or if you include multiple external style sheets and any of their rules conflict, you should know which rule and/or property's value will be used based on **basic** cascading precedence.
- You should know the different kinds of selectors we have discussed/covered/used so far:
  - These include element selectors, class selectors, id selectors, attribute selectors, and descendant selectors.
  - You should know what these are, how to write them, which HTML element(s) would be selected, and why you might want to use them.
  - You should know what it means if a rule's selector is a comma-separated list of selectors.
- You should be familiar with the properties we have discussed and used in class examples so far;
  - That said, if another property were to be described and example values for it given, you should be able to write a rule appropriately setting that property to style a selected element.
- What is the CSS Box Model? What does it describe? You should be familiar with this, and with the "pieces" within it.
  - Make sure you understand the difference between an element's padding and its margin.
- Know the basics for having an element use CSS flexbox layout.
- Know the basics for having an element use CSS grid layout, and including nested elements at desired locations within such an element.
- You should be able to center an element within another element using CSS.
- Know some ways to to format `form` elements attractively using CSS.
- Be comfortable with using the HTML `div` element; how might you use a `div` element in combination with CSS?
  - (But also be aware that HTML5 introduces new page-section elements that should be used instead of `div` elements when they semantically "fit" that content.)
- You are responsible for those CSS features that have been discussed in lecture and in lab, as well as

those CSS 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.
- How would you name a PHP-enabled file on nrs-projects? 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 are the two types of PHP tag used in CS 328? Those are the **only** types of PHP tags you are responsible for on this exam (and the **only** ones you should use on the exam!)
- 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, following CS 328 class coding standards? (note: the preferred way, in this course, is to end the regular PHP tag, and intersperse boilerplate response with PHP expression tags. That is the only approach you are responsible for.)
  - 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/interfaces that one can use with PHP to allow it to interact with databases, know that we happen to be using OCI (Oracle Call Interface) to connect PHP to Humboldt's Oracle student database.
- 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.
  - Understand how the value of `$_SERVER["REQUEST_METHOD"]` can be used to determine which of these the PHP should generate for a request.
- Be comfortable using the PHP function `array_key_exists` to see if a key exists in a PHP associative array.
  - Why, in the context of web applications, might it be useful to know if a key exists in a PHP associative array such as `$_POST` or `$_GET`?
- Make sure that you are comfortable with:
  - Knowing which superglobal associative array to use to obtain submitted values from submitted forms.
  - Obtaining submitted values from submitted forms.

  - Calling appropriate PHP functions to sanitize user inputs and/or strip tags from user inputs.
  - Connecting to an Oracle database, executing SQL statements, and retrieving results (as appropriate).
    - (This could include building a PHP that outputs the results from a SQL `select` as an HTML `table` element or as an HTML `select`/drop-down form widget.)
  - Appropriately using bind variables to create a dynamic SQL statement INSTEAD of using concatenation, to help thwart SQL injection.
    - How do you write a bind variable within a string containing a SQL statement?
    - When and how do you bind a value to a bind variable within a SQL statement?
- What is a server-side include? 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).
- 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.
  - You should understand why this is the case, also.
  - What is cross-site scripting?
  - What is SQL injection?