

CS 328 - Exam 1 Review Suggestions - Spring 2016

last modified: 2015-02-18

- 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.
- 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 `html5-template.html`
 - an example of a PL/SQL stored procedure and an example of a PL/SQL stored function
 - an example of PL/SQL exception handling
 - example HTML5 document that happens to include at least one of each of the following elements: hyperlink, form, submit button, label, fieldset, textfield, radio button, checkbox, drop-down box
 - an example of an external CSS3 file
 - an example HTML5 document using external CSS3
- 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.
- 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 the material covered regarding CSS3 basics up to the beginning of our discussion of the CSS Box Model, up to and including the Week 4 Lab Exercise and Homework 4
 - This includes element selectors, class selectors, and attribute selectors
 - The CSS Box Model and beyond will be included in Exam 2 coverage.

n-Tier Architecture

- What are the traditional components of an interactive database application?
- What do we mean by an n-tier architecture?
- What is a 1-tier architecture? a 2-tier architecture? an n-tier/3+-tier architecture?
 - what are some of the potential advantages of a 1-tier architecture over a 2-tier architecture? ...of an n-tier architecture over a 2-tier architecture?
 - on which tier are those interactive database application components typically placed in a 2-tier architecture? ...in an n-tier architecture?
 - what are the tiers in an n-tier architecture?

SQL

- Consider an n-tier architecture. On which "tier" is SQL *executed*?
- Note that you **will** be asked to write SQL queries on this exam, as they are very important in database applications. Your skills (and comfort) in writing them should be increasing during the course of this semester.
- You should be able to read a SQL query and, given example tables, determine what it would do; you should be able to modify and/or debug a SQL query.

PL/SQL

- You are expected to be able to distinguish between SQL statements, PL/SQL statements that are not also SQL statements, and SQL*Plus statements.
 - Note that PL/SQL subroutines may contain both PL/SQL and SQL statements, but **not** SQL*Plus statements.
- Consider an n-tier architecture. On which "tier" is PL/SQL *executed*?
- What does PL/SQL need to "add" to SQL, so that procedural programming will be possible?
- You are expected to be comfortable reading and writing PL/SQL stored procedures and stored functions.
 - What are the differences between a PL/SQL stored procedure and a PL/SQL stored function?
 - What are the primary goals/purpose of these?
 - What is the syntax for each?
- How must you conclude a PL/SQL stored procedure or function so that it will be compiled?
 - What SQL*Plus command should you enter to see the compilation errors for a PL/SQL stored procedure or function?
 - What SQL*Plus command should you enter to be able to see `dbms_output.put_line` output?

- Executing PL/SQL code
 - How do you call a PL/SQL stored procedure from within SQL*Plus? How do you call a PL/SQL stored procedure from within another PL/SQL stored procedure or function?
 - How do you call a PL/SQL stored function from within SQL*Plus? How do you call a PL/SQL stored function from within another PL/SQL stored procedure or function?
- You are responsible for those PL/SQL features that have been discussed in class, as well as for those PL/SQL features that have been used in posted course examples and in homeworks.
- Given PL/SQL, example tables, and example calls, you should be able to tell what would happen. Given error messages or errant behavior, you should be able to debug PL/SQL and SQL.
- What kind of parameter is a PL/SQL parameter, by default? What other two kinds of PL/SQL parameters have we discussed? How do you indicate that a parameter is one of those other kinds of parameters?
 - How can you call a PL/SQL stored procedure or stored function that expects parameters? How can you call one that does not expect any parameters?
- Make sure you know how to do the following:
 - compile a PL/SQL stored procedure or stored function
 - find the compilation errors in a stored procedure or stored function
 - execute a stored procedure or stored function
- Make sure you know:
 - the skeleton of a PL/SQL stored procedure or stored function
 - how to specify a parameter being passed to a stored procedure or stored function
 - the purpose of the `DECLARE` section (which is not always labeled that, however -- it is the section between the stored subroutine header and its opening `BEGIN`).
 - how to write a PL/SQL assignment statement
 - how to write a `SELECT` statement that puts values into program variables
 - how to use `dbms_output.put_line` (and what SQL*Plus statement is needed to be able to see its output)
 - how to use the concatenation operator for strings

PL/SQL Exception Handling

- How can you handle exceptions within your PL/SQL code?
- You should be comfortable with the syntax and semantics of PL/SQL exception handling; you should be able to read and write PL/SQL code that includes exception-handling.
- What is a pre-defined exception? What are some of the common Oracle pre-defined exceptions, and when are they raised?

- How can you raise an exception? How can you define your own exceptions (a user-defined exception)?
- You should be able to write PL/SQL code that performs desired actions when a particular exception occurs.

Intro to HTML5

- What does HTML stand for? What is HTML5? What is the intent of HTML?
- Consider an n-tier architecture. On which "tier" is HTML *executed*?
- Note that you are expected to write "strict"-style HTML5 (on exam questions as well as in homework/lab submissions).
- What is an element? What is an attribute?
- What is a block element? What is an inline element? What are examples of each? What must an inline element be contained within?
- You are responsible for those HTML5 features that have been discussed in class, as well as those HTML5 features that have been used in posted course examples and in homeworks.
 - That includes (but is not limited to) basic HTML5 document structure, as well as HTML elements such as those for titles, paragraphs, divs, spans, headings, numbered and unnumbered lists, images, hypertext links, tables, forms, fieldsets, labels, textfields, password fields, submit buttons, radio buttons, checkboxes, textareas, and drop-down boxes.
 - What is the `id` attribute? What is the particular rule with regard to its value?
 - You should also be familiar with the basic rules of "strict"-style HTML5 syntax (for example: how all tags for elements with content must be closed, the way that contentless elements must be written, how attribute values must be written, the case rules, how elements must be properly nested, etc.)
 - You should be familiar with HTML5 terminology (for example, element, attribute, tag, content, etc.) You should be comfortable with the differences between HTML5 elements, attributes of an element, and values of attributes within an element's start tag.
 - an uncommented version of the posted example `html5-template.html` will be provided in a references packet along with the exam.
- What happens when an HTML5 form is submitted? If an HTML5 page has multiple forms, you should be able to tell what happens if specified actions are taken on any of the individual forms.
 - What is the difference between a form submitted using the (default) `get` method and one submitted using the `post` method?
 - Given a form whose method is `get` (or is not specified, so that the default `get` method is used), you should be able to give the URL that would result given what has been done to the form at the time that it is submitted. (That is, what will be the `name=value` pairs within that URL?)
[BUT! I do NOT expect you to know the codes that blanks and other special characters are replaced with in those pairs within the URL!]

- If the form's method is `post`, will you still see those `name=value` pairs within the URL that would result when the form is submitted?
- Where would you normally place an HTML5 page on nrs-projects? What permissions does the HTML5 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 page?
 - How could you write a link to another HTML5 page in the same directory?
 - How could you insert an image stored in the same directory?

Intro to CSS3

- what does CSS3 stand for? What is its purpose?
- Consider an n-tier architecture. On which "tier" is CSS3 *executed*?
- what does "cascading" mean in CSS3? what does "style" mean?
- what are some of the potential benefits of CSS3?
- 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 HTML5 page use an external style sheet? (Be sure to know the CS 328-approved HTML5 element for this.)
 - What do you do (if anything) to HTML5 elements within an HTML5 page to have external CSS3 rules apply to them? (This varies based on the rule selector, of course.)
- what is the basic CSS3 rule syntax? what are its three main parts? what is the selector? a property? a value?
 - given a set of CSS3 rules, you should be able to predict how a given HTML5 document or given HTML5 elements would be displayed;
 - should be able to read, write such rules;
 - should be able to (if necessary) modify an HTML5 element to use a given 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 are not, however, responsible on the exam for the specificity scoring discussed on Chapter 4, p. 97 (that can come into play when class selectors, ID selectors, and context selectors are

involved!)

- You should know the different kinds of selectors discussed/covered/used through the Week 4 Lab Exercise and Homework 4
 - You should know what a class selector is -- how to write a class selector, and how to give an HTML5 element that class -- and why you might want to do so.
 - You should know how to write and use an attribute selector, and why you might want to do so.
- You are responsible for those CSS3 features discussed in lecture and in lab up to the Week 4 Lab, up to the CSS Box Model, as well as those CSS3 features used in posted course examples to that point and in course assignments up to Homework 4.