CIS 318 - Exam 1 Review Suggestions - Spring 2012

last modified: 2-22-12

- 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.

- 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.
  - 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).

- Your studying should include careful study of posted examples and note as well as the homeworks thus far.

SQL

- 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 comfortable with the `nvl` function -- what it does, where it can be used, and why/when you might use it.

- You should also 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.

- 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 function - can it easily be called directly from within SQL*Plus? Can it be called from another PL/SQL stored procedure or stored 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, could you tell what would happen? Given error messages or errant behavior, could you 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?

What are the restrictions (if any) on each kind of PL/SQL parameter?

How can you call a PL/SQL stored procedure or stored function that expects parameters? What are the limitations on calling a PL/SQL stored procedure with the non-default kinds of parameters?

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.
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?

Intro to XHTML

• What does HTML stand for? What is XHTML?
• You are responsible for those XHTML features that have been discussed in class, as well as those XHTML features that have been used in posted course examples and in homeworks.
  – That includes (but is not limited to) basic XHTML document structure, title, paragraphs, headings, numbered and unnumbered lists, images, hypertext links, tables, forms, text boxes, submit buttons, reset buttons, radio buttons, checkboxes, textareas, and drop-down boxes.
  – You should also be familiar with the basic rules of XHTML syntax (for example: how all tags with content must be closed, the way that contentless tags must be written, how attribute values must be written, the case rules, how tags must be properly nested, etc.)
  – You should be familiar with XHTML terminology (for example, attribute, tag, content, root tag, etc.) You should be comfortable with the differences between XHTML tags, attributes within a tag, and values of attributes within a tag.
  – an uncommented version of the posted example xhtmlcss-template.html will be provided on a references page along with the exam.
• In CIS 318, how are we neatly "lining up" or organizing the components within an XHTML form? (There are other means for doing this in CSS; we are using this XHTML container instead.) Be able to write an XHTML form using this. (Note that one often uses this same XHTML element for neatly displaying the results of a query...)
• What happens when an XHTML form is submitted? If an XHTML 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 get method and one submitted using the post method?
  – Given a form whose method is get, 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.
• Where would you normally place an XHTML page in Humboldt's set-up? What permissions does the XHTML 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 XHTML page in the same directory?
- How could you insert an image stored in the same directory?

**Intro to CSS**

- what does CSS stand for? What is its purpose?
- what does "cascading" mean in CSS? what does "style" mean?
- what are some of the potential benefits of CSS?
- How can one set up an external style sheet? and internal style sheet? an in-line style sheet?
  - in general, why are external style sheets preferable to internal and in-line style sheets? why should internal and in-line style sheets be used sparingly?
- what is the basic style syntax? what are its three main parts? what is a selector? a property? a value?
  - reading a collection of styles (possibly at different levels), you should be able to predict how a given XHTML page would be displayed;
  - should be able to read, write such styles;
  - what does a selector do/indicate? what does a property do/indicate? what does the value (in a style) do/indicate?
- if multiple levels of style sheet specify different styles, which style will be used? (that is, what is the priority level for styles at different levels? what are the 4 levels, and what is their priority level?)
- 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.

**Intro to (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 client-side JavaScript in CIS 318.
  - When we mention "JavaScript" in this class, then, you should assume that client-side JavaScript is intended unless specified otherwise
- What are some of JavaScript's capabilities?
- Consider an n-tiered architecture. On which "tier" is JavaScript executed? Be comfortable with how a document containing JavaScript is handled/processed.
- How would you name an XHTML file containing JavaScript? Where would you normally place an XHTML file containing JavaScript in Humboldt's set up? What permissions does this XHTML file need to have there? What URL would you (or an XHTML page) then use to access that XHTML file containing JavaScript?
– assume, however, that we are talking about an external JavaScript. How would you name that file? How could you include it within an XHTML file? If it contains JavaScript functions, in which part of an XHTML page would it be most appropriate to include it?

• Should be comfortable with the JavaScript syntax and features discussed in class and used in exercises and assignments (including, but not limited to):
  – how do you write and use variables in JavaScript? what is the scope of JavaScript variables?
  – how many JavaScripts can be within a page?
  – how do you write a comment?
  – how can you concatenate strings? do basic arithmetic?
  – what can you put in a page so that a browser that does not support JavaScript will at least include text warning the user of this when that page is displayed?
  – how do you write a function? call a function?
  – how do you do branching, repetition?
  – what is the difference between == and ===? What is NaN? (Note that you need to use the function isNaN to see if something has this special value.) How can you attempt to get the numeric equivalent of something?

• What is the meaning of the onload attribute of an XHTML body tag? ...of the onsubmit attribute of the XHTML form tag? ...of the onclick attribute of the XHTML button tag? How can these be used in conjunction with JavaScript?
  – note: you are expected to use unobtrusive JavaScript style

• Consider the DOM (Document Object Model) -- what is the document object?
  – How can you use its getId method to obtain a reference to an XHTML element object within that page? What attribute should an XHTML element have to allow it to work with this method?
  – How can you obtain or set the value of an XHTML element object so obtained?
  – How can you set the value of an attribute for such an XHTML element object?
  – Understand how, in the head of an XHTML 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 XHTML elements within that page.
    – What does the focus method for an XHTML element object cause to happen?

• In the context of JavaScript, what is meant by truthy and falsey?

• How can you use unobtrusive JavaScript to do simple form validation (to prevent a form from being submitted unless it is filled out "appropriately")?