

CS 318 - Exam 2 Review Suggestions - Spring 2013

last modified: 4-10-13

- 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.
- You are still responsible for SQL! (and PL/SQL, and HTML5, and CSS3, other Exam 1 topics).
 - (obviously, we are building on and making use of previous material in much of this new material.)
 - there will be at least one question focused **purely** on SQL.
 - 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.
- 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 notes as well as the homeworks (and posted homework example solutions) thus far.
- You are responsible for material covered through and including JSP, up to and including Homework 8.

Intro to Java and JDBC

- In a file with what name does a public Java class go? How do you compile a Java class from the command line? What results from compiling a Java class?
 - What is the relationship between the name of a public Java class and the name of the file its source code is in? ...and the name of the file its bytecode is in?
 - know how the Java compiler and Java Virtual Machine work together in Java translation and execution; why can Java be said to be both compiled and interpreted?
- Why is Java said to be (at least in theory) "Write once, run anywhere"?

- How can you tell if a Java class is a Java application? How can you execute a Java application?
- You are responsible for those Java and JDBC features that have been discussed in lecture and in lab, as well as for those Java and JDBC features that have been used in posted course examples and in course assignments.
 - You are responsible for knowing the Java naming and capitalization conventions discussed, also.
- What package do you need to import within a Java class that is going to make use of JDBC?
- Need to be comfortable with the basic steps required to connect to a database and run SQL queries from Java.
 - what object(s) do you need to be sure to CLOSE when you are done?
- Need to be able to tell what Java code (and applications and servlets) would do; may need to be able to modify it, debug it, or complete it. Should be able to write Java code as well.
- should know when and how to use the methods `executeQuery` and `executeUpdate`;
 - given a desired SQL statement, you should be able to say which of `executeQuery` or `executeUpdate` would be more appropriate for executing it, how you could execute it, and what that call would return.
- what is some database metadata that you can obtain using JDBC? How can you obtain the number of columns in a `ResultSet`? The label of a particular column in a `ResultSet`?
- in addition to the `Statement` class, you should also be comfortable with `PreparedStatement` and `CallableStatement`;
 - what is the difference between how you set up and execute a `PreparedStatement` as compared to a `Statement`? What must be done before each call of a `PreparedStatement`?
 - should know two major advantages of using a `PreparedStatement` over a `Statement`. When would you choose to use a `PreparedStatement` instead? When would it not matter so much?
 - when would you use a `CallableStatement`? How do you set it up? How do you obtain the results from this?
- you should be able to call a PL/SQL stored procedure or function using JDBC;

Java servlets

- What is a Java servlet? Where is a Java servlet executed, in terms of an n-tiered architecture? How can you tell if a Java class is a Java servlet?
 - what packages should a servlet import?
- What is/can be the purpose of a servlet? What actually calls it? Need to be comfortable with how a Java servlet can "fit" in an n-tiered architecture.
- what are some aspects that help with servlet efficiency?
- Need to be comfortable with how a Java servlet is "set up", compiled, and executed here at HSU. I could ask questions directly addressing this.
- You are responsible for those servlet features that have been discussed in lecture and in lab, as well as those servlet features that have been used in posted course examples and in course assignments.

- where (basically, not specifically) would you normally place a Java servlet in Humboldt's set-up? What file would you put there? What URL would you (or an HTML page) then use to access that servlet?
- Need to be able to tell what Java servlet code would do; may need to be able to modify it, debug it, or complete it. Should be able to write Java servlets as well.
- what happens, at the web server level, when a page requests a Java servlet?
- when is a servlet's `doGet` method called? when is a servlet's `doPost` method called?
- what are the two parameters to the methods `doGet` and `doPost`? What do they represent? How can they be used?
- within the servlet code, what do you need to do to craft an HTML response?
- should be comfortable with how HTML forms can interact with Java servlets; should be comfortable with servlets that can be either directly accessed via their URL or that are reached from an HTML form.
 - which servlet method is called in each case?
 - given an HTML form, how do you indicate that the form has a servlet as its action? How can you tell which servlet method will be used to handle that form, when it is submitted?
 - make sure that you are comfortable, within a servlet, with obtaining parameter values from form inputs.
- you should be able to call a PL/SQL stored procedure or function from a servlet;

Finite state machines (FSMs)

- (also known as finite automata, deterministic finite state machines, deterministic finite automata)
- could give you a "classic" FSM, and ask you if given strings are accepted by that FSM; could ask you to name its start state, its accepting states; could ask you, given the current state and an input, what the resulting state/next state would be;
- you should be able to read and answer questions about a FSM representing a simple web application as well (in the style discussed in class and as used in several homework handouts)
- I will **not** ask you to draw a FSM for Exam 2 (maybe on the Final...)
- (you are only responsible for the "classic" FSM notation discussed in lecture; you are not responsible for UML-style FSMs.)

Java session management

- what is the nature of HTTP with regard to state? Given **just** HTTP (that is, I'm not talking about servlets or JSP or cookies or other features external to HTTP), can you associate a request with a previous request?
- you should know that Java creates a session object for each browser session, and that this session object can have attributes added to and retrieved from it by any servlet/JSP run during the session;
- within a Java servlet, with what method can you obtain the current `HttpSession` associated with a request or return a new `HttpSession`? What object has this method? (You'd call this method for what object?) What is the restriction for when you can/should call this method?
 - note that the above is done for you with JSP; you can simply use the resulting pre-defined

`HttpSession` object, which is named `session`

- know the three ways that a session can be terminated:
 - using the `HttpSession invalidate` method
 - the session can expire
 - the *server* can be restarted
 - NOTE that "closing the browser" is NOT one of these! The user closing his/her browser window does **not** terminate the session!
- you should be comfortable with the `HttpSession` methods discussed and demonstrated; these include methods for getting the current session ID, setting a session attribute, getting a session attribute, determining whether the client yet knows about the session (`isNew`), and invalidating the session (including unbinding objects bound to the session)
- Need to be able to tell what code involving sessions would do; may need to be able to modify it, debug it, or complete it. Should be able to write code involving sessions as well.
- consider the types that `HttpSession` methods `getAttribute` and `setAttribute` expect as arguments and return as their result.
 - what does this imply about what kinds of values can be session attribute names? What does this imply about what kinds of values can be session attribute values?
 - when obtaining the session attribute value, what might you have to do to the obtained value in order to, for example, assign it to a variable?

JSP - JavaServer Pages

- what does JSP stand for? What is a JSP? What does it contain? What happens when a client requests a JSP? What happens if there are successive requests to a JSP?
- Like JavaScript, JSP involves embedding executable code within HTML. You should be able to understand and compare/contrast the similarities and differences between JavaScript and JSP.
 - what is the suffix of a file containing HTML and JSP? of a file containing HTML and JavaScript?
 - where (in terms of an n-tiered architecture) are JSP code snippets within an HTML page actually executed? where are JavaScript code snippets within an HTML page actually executed (using the style of JavaScript assumed in this course)?
 - if an end-user/client were to view the "page source" of an HTML page involving JSP, what would he/she see? ...if he/she were to inspect the "page source" of an HTML page involving JavaScript, what would he/she see (using the style of JavaScript assumed in this course)?
- what is the expected suffix for a JSP?
- What are the kinds of JSP tags that we have discussed? You should be able to write examples of different JSP tags; you should be able to identify the different kinds of JSP tags.
- You are responsible for those JSP features that have been discussed in lecture and in lab, as well as those JSP features that have been used in posted course examples and in course assignments.
- Need to be able to tell what JSP code would do; may need to be able to modify it, debug it, or complete it.

Should be able to write JSPs as well.

- you should know of at least three pre-defined objects available to each JSP, and how to make use of them;
- what is the relationship between JSP and Java servlets? What happens to a JSP when it is accessed?
- recall the advantages and disadvantages of Java servlets -- do these also apply to JSP?
 - how many servlets are created per JSP? How many HTML pages may a "hand-written"/non-JSP-created Java servlet create?