Course Syllabus for CS 318
Programming Database Applications
Spring 2013

Basic Course Information:

Instructor:	Sharon Tuttle		
Lecture time and location:	Wednesday	1:00 – 2:50 pm	MUSA 130
Lab time and location:	Monday	1:00 - 2:50 pm	BSS 313
Instructor's office:	BSS 322		
Instructor's e-mail:	st10@humboldt.edu or sharon.tuttle@humboldt.edu or smtuttle@humboldt.edu		
Instructor's office phone:	(707) 826-3381		
Instructor's office hours:	Monday, Wednesday	3:30 - 5:00 pm	
	Tuesday	1:30 - 2:30 pm	
	Friday	4:00 - 5:00 pm	
	or by appointment		
Course public web page:	follow link from: http://users.humboldt.edu/smtuttle/ or follow link from course Moodle site		

Course Description:

The focus of this course is database programming, that is, the development of applications using a relational database as the primary source and sink of data. We will look at the development of such applications from several perspectives: standalone PL/SQL programs (stored procedures, stored functions, and triggers); using Java servlets and JDBC; JSP (Java Server Pages); and PHP. We will also make use of JavaScript, XHTML, and CSS on the client-side. This is primarily a programming class.

(Note that I am assuming that those taking this course are very comfortable with SQL and C++, and have just an introductory knowledge of PL/SQL and PL/SQL triggers; prior knowledge of Java, PHP, XHTML, CSS, and Javascript is **not** assumed.)

However, CS 318 is distinguished from language-specific programming classes (such as C++ or Java) in that it is less focused on literacy in particular programming languages (although that will be a desirable side-effect) and more focused on concepts and considerations surrounding database programming. The hands-on portion of the course is used to anchor and exercise the conceptual components. Even though your energies will tend to be dedicated to getting particular programs to behave in particular ways according to the syntactical rules of the software platform, you should always simultaneously seek to identify how you are engaging in special cases of the more general.

Course Objectives:

After successfully completing this course, students should be able to: *

- design and implement usable and accessible web applications that use a database as their sink and source of data;
- understand the differences between the tiers within an n-tiered architecture and the tradeoffs in components of a system being placed within those different tiers;
- design and implement effective SQL queries to be embedded within database applications to query and manipulate database data effectively;
- illustrate how interactive client-server web applications that interact with a database can be built using several different types of Web technologies;
- demonstrate how to implement a database-driven web site, explaining the relevant technologies involved in each tier of the architecture and the accompanying performance tradeoffs.

CS Program Learning Outcomes that this course addresses:

This course addresses departmental learning outcomes of:

- Computational Thinking
- Self-Directed Learning
- Technical Writing
- Communicating and Collaborating

This course addresses computational thinking at a moderate to advanced level, adding the concept of applications using n-tiered architectures. It addresses self-directed learning at a moderate level by encouraging students to practice searching software documentation to expand their skills beyond the quick introductions to the various languages used in this course. It addresses technical writing and communicating at a moderate level via the topics of design of the presentation tier and design for usability.

CS Program Learning Outcomes that this course addresses:

This course explicitly contributes to students' acquisition of skills and knowledge relevant to HSU Learning Outcomes:

HSU graduates will have demonstrated:

- Critical and creative thinking skills in acquiring a broad base of knowledge and applying it to complex issues.
- Competence in a major area of study.

HSU graduates will be prepared to:

* Some of these are adapted from the ACM Computer Science Curriculum 2001, available from link at: http://www.acm.org/education/curricula-recommendations

- Succeed in their chosen careers.
- Take responsibility for identifying personal goals and practicing lifelong learning.

Course Prerequisites:

CS/CIS 315 - Database Design & Implementation and Math 253 - Discrete Mathematics,

or instructor approval.

Required Course Text, Materials, etc.:

- "Web Programming Step by Step", Second Edition, Marty Stepp, Jessica Miller, Victoria Kirst, Lulu, 2012.
  - authors' official companion web site: http://www.webstepbook.com/
- Turning Point RF Response Clicker, available at the campus bookstore
- Additional required readings will be made available either on-line, or via resources available through the HSU Library such as the ACM Digital Library and Safari TechBooks Online.

Course Software:

We primarily will be using an **Oracle implementation of SQL** for this course. Unless noted otherwise, you are expected to use the on-campus `student` Oracle database for course assignments, and exam questions involving SQL will be assumed to use Oracle SQL as well. The software you use for any additional practice is, of course, up to you.

Throughout the semester, you will be making some use of the UNIX operating system. Note that you may access nrs-labs.humboldt.edu and nrs-projects.humboldt.edu by using `ssh` (secure shell) and `sftp` (secure ftp); one of several versions of ssh may be downloaded for free from:

http://www.humboldt.edu/its/software

(A nice introductory tutorial, including screenshots, for the graphical Windows version of `ssh` and `sftp` can be found at:

http://oit.colorado.edu/node/1657

)

You should also be able to use the free **SQL Developer** software available from Oracle, which has versions for Windows, Linux, and Mac OS X, and can be downloaded for free from:

http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html

Oracle PL/SQL is available via your Oracle `student` database account. We will be using Java mostly in a server-side fashion, so that Java's JDBC can connect to the Oracle `student` database, but a variety of free Java IDE's are also available to make program entry more convenient -- for example, DrJava (http://www.drjava.org/), NetBeans (http://netbeans.org/), and Eclipse (http://www.eclipse.org/). We should be using either XHTML or HTML5, CSS or CSS3, JavaScript, JSP, and PHP as well.

Clickers:

We will be using Turning Technologies student response clickers in class. There is significant literature indicating that using clickers may increase student engagement and success in learning.

Students purchase this clicker (they are available from the campus bookstore); purchased clickers can be returned at the end of the semester for a partial refund of the purchase price or they can be kept for use in future HSU courses. You register your clicker at the beginning of each semester by logging into Moodle and going to the course Moodle site; you should then be asked to enter your clicker code (consisting of 6 characters/digits) from the back of your clicker when requested. You then bring your clicker to every class meeting.

These clickers will be used for in-class questions, which will be interspersed within class sessions. The response system will record the overall class response percentages as well as keep track of individual answers. Typically, you will receive **2 points** for a correct answer, **1 point** for an incorrect answer, and **0 points** for no answer, but with a maximum semester clicker-questions grade of **120**. (There may be some no-point questions from time-to-time as well -- such questions will be noted if/when they come up.) Thus you will be rewarded for regular attendance and participation. If you miss a class session, you miss that day's clicker questions and cannot make them up. However, there will be a sufficient number of questions asked to allow for the possibility of extra credit (up to a **maximum** clicker grade of **120**) or to make up for a day that you are out due to illness (although note that you are still responsible for finding out what you missed on such days).

If you forget your clicker for a class meeting, then **up to 5 times** you may still receive some clicker credit, **minus a 2-point penalty**, by e-mailing me your clicker answers for that day, **by midnight on that day**, using a Subject: line of: `Subject: CS 318 Clicker Answers for <date>`. Later e-mails, or e-mails without the proper `Subject:` line, will not be accepted for credit.

The idea is that the clicker questions will help you to see if you are starting to understand concepts being discussed; sometimes they will also provide review of concepts discussed previously. Clicker questions are typically quite different from exam questions (since clicker questions are typically multiple-choice questions, while exam questions will rarely be multiple-choice). They still enable you to get some immediate feedback regarding whether you are grasping course concepts, whether you need to pay more attention to course discussions and/or readings, etc. They may even help me to know what concepts might need more explanation in-class.

I hope to run tests of the system during the first two class meetings, and hope to begin asking questions that "count" during the second week's Wednesday lecture. Therefore, you must purchase your clicker and register it as soon as possible. If there is an issue with this (for example, if the bookstore runs out of clickers), contact me immediately.

Finally, please note that use of another CS 318 student's clicker, or having someone else use your clicker in a CS 318 class session -- that is, pretending that someone is in class who actually is not -- is considered to be cheating, with the same policies applying as would be the case if you turned in someone else's work as your own or permitted someone else to copy your work. Please ASK ME if you are not sure what I mean by this.

Grading Breakdown:

Your semester grade will be determined by the percentage of points that you earn, **subject to some minimum requirements**. Here are the grade percentages, followed by those minimum requirements:

Homework assignments:		25.0%	
Lab exercises:		12.5%	
Clicker questions:		12.5%	
Exams:	**Exam 1:**	15.0%	
	Exam 2:	15.0%	
	Final Exam:	20.0%	Monday, May 13, 12:40 - 2:30 pm, MUSA 130

Grade Requirements:

1. To earn a grade of **C or better** in this course, the following three requirements must **all** be met:
   - your overall semester average must **equal or exceed 72.5%** - this is to show a reasonable level of overall mastery of the course material.
   - the **average** of your Exam 1, Exam 2, and Final Exam grades must **equal or exceed 60%** - this is to show

that you understand at least a minimal reasonable level of the most important course concepts.

- the **average** of your Homework assignments must **equal or exceed 60% -** because this is a programming course, but programming acumen is not tested as effectively on exams, this is to show at least a minimal level of database programming competence and experience in addition to course concept mastery. Also, past experience has shown that students who do not put a solid effort into the course homework assignments do not do well on the course exams.

2. If **all three** requirements above are **not** met, then your semester grade will be **either C-** or the letter grade computed according to the mapping given below, **whichever is lower**.
   - (That is, if a student had an overall semester average of 74% but a Homeworks average of 55%, that student would receive a **C-** for his/her semester grade; if a student had a Homeworks average of 61% and an Exams average of 71%, but an overall semester average of 65%. then that student would receive a **D** for his/her semester grade. You are expected to ASK ME if this aspect of the grading policy is not clear to you.)
3. Including the three requirements noted above, your semester grade will be computed according to the mapping given below:

Overall Percentage (based on the given weights)	**Exams Average**	**Homework Average**	**Letter Grade**
>= 93	>= 60	>= 60	A
>= 90 and < 93	>= 60	>= 60	A-
>= 87 and < 90	>= 60	>= 60	B+
>= 83 and < 87	>= 60	>= 60	B
>= 80 and < 83	>= 60	>= 60	B-
>= 77 and < 80	>= 60	>= 60	C+
>= 73 and < 77	>= 60	>= 60	C
>= 73	< 60	any	C-
>= 73	any	< 60	C-
>= 70 and < 73	any	any	C-
>= 67 and < 70	any	any	D+
>= 60 and < 67	any	any	D
< 60	any	any	F

Final Exam:

Again, the Final Exam for this course is scheduled for **Monday, May 13, 12:40 - 2:30 pm**, in **MUSA 130** (unless I announce otherwise). Note this time and date BEFORE making your end-of-semester travel plans.

Students with Disabilities:

Persons who wish to request disability-related accommodations should contact the **Student Disability Resource Center** in the Learning Commons of the Lower Library, **826-4678 (voice)** or **826-5392 (TDD)**. You can reach the Student Disability Resource Center's web site at:

http://www.humboldt.edu/disability/

Please note that some accommodations may take up to several weeks to arrange. If you are eligible for such accommodations, please contact me as soon as possible to discuss them.

Add/Drop Policy:

Students are responsible for knowing the University policy, procedures, and schedule for dropping or adding classes. You can find these on the web at:

http://www.humboldt.edu/registrar/students/regulations/schedadjust.html

You can find the University policies for repeating classes at:

http://www.humboldt.edu/registrar/students/regulations/repeat.html

Note that the CSU (and thus HSU) policies on withdrawing from and repeating courses changed as of Fall 2009:

- Students may withdraw from no more than 18 semester-units after the first four weeks of instruction; that is, students may withdraw from no more than 18 semester-units between census and the final 20% of instruction, and only then with a serious and compelling reason. (Note that: "Withdrawal from courses for reasons that are catastrophic, such as accident or serious illness do not count toward the 18-unit limit." [from the Registrar's web site])
- Students may repeat courses for grade forgiveness only if they earned grades lower than a C.
- Students may repeat up to 16 semester-units with grade forgiveness.
- Students may only repeat a course for grade forgiveness two times and each of these attempts counts toward the 16-unit maximum for repeats.
- Students may repeat up to an additional 12 semester-units with grades averaged.

Please note - as of Fall 2009, HSU is being much more strict about what constitutes a "serious and compelling reason".

The census date for Spring 2013 (before which you can drop without a W, and without it counting toward your 18 semester-units drop limit) is: **Monday, February 18th**.

The last date for Spring 2013 to drop with a W on your transcript, with a serious and compelling reason, and subject to the 18 semester-unit drop limit, is: **Tuesday, April 9th**.

If you do drop the course, note that it is **your responsibility** to complete and submit the appropriate forms.

Incompletes:

Incompletes are rarely given and only in the case of a true emergency. They certainly are not appropriate for students who find they have fallen behind on assignments, missed a test, or taken on too much academic, work, or family responsibilities. For these situations, dropping the course would be appropriate (**if** that is still possible according to the University policies for dropping courses).

Course Expectations:

First: remember the general rule of thumb for college-level courses:

To be successful in a course, you should plan to spend at least 3 hours outside of class for each 1 hour of college course credit. That implies an estimate of **at least 9 hours a week spent outside of class for this 3-credit course**.

However, you should be aware that:

- This is a junior-level CS major course; it has an accordingly-rigorous workload. It involves a large amount of programming at different tiers.

- This is a programming course; you cannot start learning the different languages we will be using at different tiers without practice!
- Programming courses can be notorious time eaters. Occasionally, a problem with code will take large amounts of time to locate and fix.
  - Starting **early** enough so that you have time to ask me questions when you run into problems can help with this...
  - ...why spend 4 hours struggling with a frustrating roadblock the night before the assignment is due, when you can spend 10 minutes composing an e-mail early in the week, work on other problems while waiting for the answer, and then get a reply that makes everything clearer as soon as you read it?
- You can only learn how to program the different tiers of an n-tiered database application by practicing doing so, and it takes some much longer than others to master this. Practicing as much as possible helps.
  - (This means playing around with in-class examples, experimenting to see if something you are curious about really works like you think, doing further research on topics of interest, and so on.)
- Later concepts are built upon earlier concepts as the course progresses -- if you ask me as soon as you realize that some concept is not clear to you, that can help keep you from falling behind.
- Homework deadlines will **not** be extended because you waited too late to start or because you did not allocate enough time before the deadline to work on it; likewise, they will **not** be extended because of hardware or network failure. (Admittedly, campus failures might affect deadlines. But don't assume so until you have heard from me definitively.) You need to keep backups of your files at all times, and need to plan your schedule to be able to work on on-campus computers as necessary.
- If you have not completed an assignment by the deadline, **your best choice is to submit whatever you have managed to do by then**, as partial credit is your friend, to carefully study the posted example solution as soon as it is available, to ask me about anything there that is still unclear, and to get a good early start on the next homework.

A successful student in this class will:

- Attend every lecture and lab, clicker at the ready.
- Participate in class (discussing clicker answers with other students, asking questions, paying attention, taking notes, being an attentive partner when pair-programming in lab).
- Complete reading assignments in a timely fashion.
- Practice and "play around" with posted examples.
- Ask specific questions -- in class, in lab, in office-hours, and in e-mail.
- Read through each homework assignment as soon as it is posted.
- Start working on each homework assignment early in the week.
- E-mail the instructor with specific homework-related questions starting early in the week both to clarify what a question is asking for and when hitting roadblocks (being sure to include both the code involved and any error messages or descriptions of bizarre behavior).
- Always submit SOMETHING for a homework assignment, even if it is not complete. (I believe in partial credit on homeworks, believing that if you have at least started working on a problem, the posted example solution will be more helpful/understandable than if you have not.)
- Compare his/her homework solutions to posted example solutions when they become available.
- Study with others for exams, practice explaining concepts to one another.

- Attempt every exam problem, and carefully study over exams when they are returned.
- Practice programming at the different tiers as much as possible.

Academic Honesty:

Students are responsible for knowing policy regarding academic honesty. For more information, visit:

http://www.humboldt.edu/studentrights/academic_honesty.php

Observe that among the actions that are unacceptable are submitting another's program, code, or file as your own and failing to quote material taken from another person's written work. (Note that copying another student's comments is also unacceptable.)

All course work is to be the work of each student, **individually**, **unless** it is **explicitly** stated otherwise at the beginning of that course work's description. Except for explicit exceptions, this is **not** a group or team programming course. If group work is explicitly permitted for some assignment, the names of all students involved must be included on the work submitted. (For example, if **pair programming** is explicitly specified as being allowed for an assignment, then each pair-programmed file turned in will include both of the names of the students who worked on it as a pair.)

(**Important aside**: pair programming specifically means that two people sit at one computer, with one typing while the other says what to type. Both people are actively involved in the programming process. Pair-programming is **not** two people working at two computers, each doing different parts of the work individually. If pair-programming is ever explicitly permitted, then you are expected to actually pair-program any files you do not complete on your own.)

(**If** an assignment does explicitly specify that it is acceptable to pair program or work in groups, make sure that you don't get into the situation where you are merely watching someone else learn.)

For homework assignments (that are not explicitly specified as permitting pair-programming), students may discuss general approaches **as long as no one involved in the discussion is writing anything down or typing anything during such discussions**. Students may also help one another in determining causes of program bugs, or in determining the meaning of compiler error messages. However, in general, students may not work together to complete homework assignments, one student should not instruct another in how to write the code for a homework assignment, and **any type of copying or modifying of another person's computer files, OR of providing computer files to another, related to homework assignments is definitely over the line, and never justified**. This applies to copying of documentation and comments as well as to copying of program code.

Note that it is **your** responsibility to ensure that course assignment files are read-protected. If you are careless about this, and someone else copies your work, you will share the penalty. (In particular, be very careful about leaving work on shared network drives in campus labs, or in UNIX/Linux directories that are not read-protected.)

Learning takes hard work; when students turn in others' work as their own, it is a slap in the face to those seriously interested in learning. Not turning in an assignment results in no credit for that assignment, of course, but that is an honest grade. Work that violates the course honesty policy deserves a lower grade than that, and therefore the course policy is that work violating this policy will receive **negative** credit. A person providing a file for copying receives the same **negative** credit as the copier. Repeat offenses will be handled according to University policies.

Asking Questions/Getting Help:

- Include `CS 318` along with the subject of your e-mail in the `Subject:` line of any class-related e-mail that you send me. This will help your e-mail be more recognizable as a class-related message, and will make it less likely that I will accidentally overlook it.
  - That said, if I have not replied to your e-mail within 24 hours, please re-send it, just in case I did overlook

it somehow.

- Also, DON'T INCLUDE the word "`password`" in your e-mails to me -- `pwd` is a handy abbreviation to use instead -- because, due to phishing scams, HSU's spam filtering does not seem to like e-mails with that word in it! (Odd, but this was definitely the case in Spring 2010...)

- I try to check my e-mail (`st10@humboldt.edu` or `sharon.tuttle@humboldt.edu` or `smtuttle@humboldt.edu`) about once a day on weekdays, and about once over each weekend. This is another reason to start assignments early, so that you have time to receive a reply to any questions that might arise.

- You are encouraged to ask me questions in class, in office hours, and by e-mail. The most successful students are those who are not afraid to ask questions early and often (I will gently let you know if you are overdoing it), who do the assigned reading, who attend lecture and lab regularly, who start homeworks promptly after they are made available from the course web page, and who practice course concepts as much as possible.
  - It is better to ask a question sooner than later -- for example, it is better to send an e-mail with a specific question as soon as you think of it than it is to wait a day or two until the next class meeting or office hour. If you wait to ask such questions, you may not have time to complete the assignment.
  - It is perfectly reasonable if you send me a question and then end up finding out the answer yourself before you receive my answer; likewise, it is not a problem if you end up sending me several questions in separate e-mails (as you work on different parts of a homework while awaiting earlier answers).

- That said, I am expecting that you will ask **specific** questions – overly vague or broad questions are problematic.
  - (For example, an example of a specific question is, "When I try to run the query: (paste in the query), I receive the following error message: (paste in the error message) Can you point me in the right direction about what is wrong?" An example of an overly vague or broad question is: "Here's my SQL script/PHP file/XHTML/CSS/etc. Is it right?")

Additional Coursework-Related Policies:

- You should not expect to be able to finish homework assignments during the lab sessions -- although you may occasionally get some lab time to work on homework assignments, typical lab sessions will include a lab exercise that is to be completed in lab. Even when you finish the lab exercise early, it will still be the case that, like any college-level course, you should expect to put in a significant amount of time outside of scheduled class meetings (lectures and labs) doing the assigned reading, working on homework assignments, and practicing concepts discussed.

- Each assignment must be submitted as is specified on its handout to be accepted for credit. This may vary for different assignments. Often, parts of assignments will be submitted using a special tool on nrs-projects.

- Each assignment will be clearly marked with one or more due dates (a single assignment could have multiple parts with multiple due dates).
  - **No assignments will be accepted late. If you wish to receive any credit for an assignment, then you must turn in whatever you have done, even if it is incomplete, by the deadline. Partial credit is usually preferable to no credit.** Note that "the computer/network/etc. going down" is no excuse --- if you leave an assignment for the last minute and there are technical problems, you still must turn in whatever you have by the deadline. As with any work done on computer, make frequent back-ups of your files!
  - You may submit multiple versions of assignment files before the deadline; I will grade the latest pre-deadline submission unless you inform me otherwise. This is to encourage you to turn assignment parts in early (since you will know that you can always turn in an improved version if further inspiration strikes). You also don't have to worry about forgetting to submit something that has already been submitted.

  - If for any reason you cannot submit course work using the submission tool on nrs-projects, e-mail me your homework files as attachments by the deadline, and then submit the files using the submission tool as soon as you are able. The e-mailed files will establish that these files were completed by the deadline.
- The tool that you will be using to submit assignments results in a file that serves as your "receipt" for having submitted items. You are expected to retain these "receipt" files at least until a grade has been posted to the course Moodle site for that assignment. If there is a system glitch or other hardware/software/network problem, you may be asked to make me a copy of one or more receipt files; if you do not have them, then you will not receive credit for the files involved. These receipt files are for your protection!
- It is nearly impossible to write unambiguous specifications. If you have questions about "what she means", get them resolved very early in the development cycle by **asking**.
- There is more to computer code than simply whether it runs or not...
  - Part of your grade will be determined by how well your work meets the written requirements. Work that you turn in is expected to meet handout specifications precisely; when one eventually works within a team on large projects, following the specifications precisely is vital, and can mean the difference between a working product and one that just sits there.
  - Note that work may be graded on **style** as well as on whether it runs properly and whether it precisely meets the homework specifications and requirements. Discussions on style will be ongoing throughout the semester.
- Some course work may be graded simply based on whether it has been attempted (the instructor's decision is final as to whether this is the case) -- other course work may be graded for correctness, style, and whether it meets specifications. You will not know in advance which will be the case.

Additional Grading-Related Policies:

- Clicker questions will be given during most lectures and labs (although probably more clicker questions will be asked during lecture sessions); graded lab exercises will be given during most lab sessions.
  - The **two lowest lab exercise grades** will be dropped from your grade.
  - Between the ample quantity of clicker questions and the dropped lab exercise grades, then, you can be absent several times from non-exam lecture or lab sessions without direct penalty, for whatever reason (although you are, of course, still responsible for the material covered on those days, and it is **your responsibility** to determine what that material is).
- Note: **NO** homework grades are dropped; **ALL** homework grades count toward your homework average. Every homework includes important practice of course fundamentals.

Additional Course Policies:

- You are expected to read this syllabus and be prepared to sign a statement that says you have received it, have read it, and understand its contents.
- Exam dates are given in the course schedule below. Make-up exams are only possible by special prior arrangement or because of a valid medical excuse.
- You should monitor your e-mail for course-related messages. The University provides a means for you to specify your preferred e-mail address, so if you wish to receive e-mail into an account other than the one HSU provides, change your preferred e-mail address in both Account Center and Moodle accordingly. Course-related messages from me will include `CS 318` in the `Subject:` line.
- You are expected to check the public course web page and the course Moodle site regularly --- course handouts, homework assignments, examples from lectures and labs, and possibly more will be posted to the

public course web page, and grades will be posted to the course Moodle site. You are expected to monitor your posted grades and let me know about any discrepancies.

- When reading assignments are given, you are expected to prepare (read and study) assigned readings before class and to participate in class discussions. Projected examples will be utilized frequently during discussion. You should understand that there may be material in the reading that will not be discussed in lecture/lab, and material in the lectures/labs that may not be found in the reading. You are responsible for both.
- **Attendance and disruptive behavior:** Students are responsible for knowing policy regarding attendance and disruptive behavior:

  http://www.humboldt.edu/studentrights/attendance_behavior.php
- Regular attendance at lecture and lab sessions is expected. If you should happen to miss a lecture or a lab, then you are responsible for finding out what you missed. "I wasn't there that time" is never an acceptable excuse. Lecture and lab notes are not posted, although many of the projected examples will be made available on the public course web site. Clicker questions and graded lab exercises missed **cannot** be made up later.
- As previously mentioned, during lab sessions, there may be lab exercises due during that lab session. Once a lab's lab exercise is complete, the remaining lab time should be used to continue work on the current course homework assignment, to practice course concepts, and/or to ask questions about course-related topics.
- **Late arrival to class:** Please attempt to come to class on time, with your headphones put away and your cell phones turned off. If you must arrive late or leave early, please do so with the least possible distraction to other students. If your late/early habits become disruptive, you may be asked to leave the class permanently.
- **Class disruption:** University policy requires that instructors eliminate disruptions to the educational process. Distractions such as excess talking, ringing cell phones, working on assignments for other classes, inappropriate or distracting laptop/tablet/smartphone/gadget use, demonstrations of affection, packing of books early, loud music leaking from headphones, chronic late arrivals or early departures, excessive comings and goings or other behaviors that disrupt the class are not acceptable. Students indulging in such behaviors will first be warned before being required to leave the class permanently.
- **Emergency Evacuation:** Please review the evacuation plan for the classroom (posted on the orange signs), and review the campus Emergency Preparedness web site at:

  http://www.humboldt.edu/emergencymgmtprogram/campus_emergency_preparedness.php

  ...for information on campus Emergency Procedures. During an emergency, information regarding campus conditions can be found at **826-INFO** or:

  http://www.humboldt.edu/emergency

Tentative Course Schedule: (subject to change!)

Note also that **additional readings are likely to be added** to those given below.

Week 1: January 23

- (January 16th was the Martin Luther King, Jr., Holiday)
- Topics: Intro to course; Intro to n-tiered architectures

Week 2: January 28, 30

- Topics: DB-SERVER-SIDE: PL/SQL triggers, stored procedures, stored functions, and exception-handling
- Homework 1 out

Week 3: February 4, 6

- Reading: Course text: Chapter 2, Chapter 6 sections 6.1, 6.2
- Topics: CLIENT-SIDE: intro to XHTML or HTML5, including forms and tables
- Homework 1 due, Homework 2 out

Week 4: February 11, 13

- Reading: Course text: Chapter 3, maybe parts of Chapter 4
- Topics: CLIENT-SIDE: more on XHTML or HTML5, intro to CSS or CSS3
- Homework 2 due, Homework 3 out

Week 5: February 18, 20

- Reading: Course text: parts of Chapter 8 and Chapter 9
- Topics: CLIENT-SIDE: intro to client-side JavaScript
- Homework 3 due, Homework 4 out

Week 6: February 25, 27

- Topics: working toward WEB-SERVER-SIDE: intro to Java-needed-for-CS-318 (intro to Java and JDBC), review for Exam 1
- Homework 4 due

Week 7: March 4, 6

- **Monday, March 4: Exam 1**
- Wednesday, March 6: NO CLASS - Instructor at conference
- Homework 5 out

Week 8: March 11, 13

- Topics: working toward WEB-SERVER-SIDE: intro to Java and JDBC
- Homework 5 due, Homework 6 out

Spring Break - March 18-22

Week 9: March 25, 27

- Topics: WEB-SERVER-SIDE: Intro to Java servlets
- Homework 6 due, Homework 7 out

Week 10: April 3

- Monday, April 1: NO CLASS - Cesar Chavez Holiday
- Topics: WEB-SERVER-SIDE: reminder of finite state machines (FSM's), discussion of the statelessness of HTTP, and intro to Java sessions
- Homework 7 due, Homework 8 out

Week 11: April 8, 10

- Reading: Course text: Chapter 15
- Topics: WEB-SERVER-SIDE: A few words on security: XSS (cross-site scripting) and SQL injection; Review for Exam 2
- Homework 8 due

Week 12: April 15, 17

- Monday, April 15: **Exam 2**
- Topics: WEB-SERVER-SIDE: intro to JSP
- Homework 9 out

Week 13: April 22, 24

- Reading: Course text: Chapter 5, Chapter 6 section 6.4
- Topics: WEB-SERVER-SIDE: Intro to PHP
- Homework 9 due, Homework 10 out

Week 14: April 29, May 1

- Reading: Course text: Chapter 7, Chapter 12 section 3
- Topics: CLIENT-SIDE: Intro to Web Design and Usability, WEB-SERVER-SIDE: Intro to XML
- Homework 10 due, Homework 11 out

Week 15: May 6, 8

- Reading: Course text: Chapter 12, sections 12.1, 12.2
- Topics: a few words on AJAX; presentations/peer critiques of web design/usability problem(s) from Homework 11; review for Final Exam
- Homework 11 due

Final Exam:

MONDAY, May 13, 12:40 - 2:30 pm, in MUSA 130 (unless I announce otherwise)