CIS 3308 Web Application Programming Syllabus
(Upper Level CS Elective)

Course Description

This course introduces the techniques used in the design and implementation of web applications, focusing on server side programming. Using development tools such as Netbeans and MySQL Workbench, students will write code for the browser (HTML, CSS, javascript/jquery, ajax), web application server software (java/JSP), and database (SQL). Each student will program their own web application as they complete the weekly lab assignments.

This course covers internet protocols (http, https) and related server side web application objects (request, response, and session) that enable us to write code that (1) extracts user entered data and (2) implements logon/authentication and authorization. Once we learn/review basic database concepts, students create their own mySql database and populate them with data. Design patterns such as MVC (Model-View-Controller) are employed to create multi-tiered, distributed web application software. Students will learn about XML and JSON (formats commonly used to transfer data over the internet) and will use ajax – a technique that uses javascript/jquery to make asynchronous HTTP requests, then receive data to modify the web page (dynamic HTML). Students will learn that ajax is a great way to take advantage of Web APIs offered by others.

Prerequisites

- Grade of C- or better in CIS 2107 Computer Systems and Low-Level Programming
- Grade of C- or better in CIS 2168 Data Structures

The only assumption is that all students have knowledge of the topics covered in the pre-requisite courses listed above. All other topics will be introduced as new material, even though some students may already have had some exposure. Students with more experience can add extra functionality to their weekly lab assignments (if they wish), as long as they meet all requirements and submit on time.

Textbook: There is no text book. Web references and other materials will be posted in blackboard.

Learning Objectives

By the end of this course, each student should be able to:

- Create a web application and using these languages: HTML, CSS, javascript, java/JSP, and SQL.
- Use MySql Workbench (an open source graphical front end to MySql) to create a normalized database, populate it with data, write SQL select statements that join database tables.
- Understand how database constraints (like primary key, foreign key) affect a web application. Write code that prevents database errors and/or appropriately handles database exceptions.
- Describe the benefits of new technologies and employ techniques such as jquery (commonly used javascript function library that provides client-side/browser animations) and ajax (partial page refresh technique, being able to utilize web APIs provided by others).
- Understand and employ good programming practices that result in reliable and maintainable web applications, e.g., MVC software design pattern, Single Responsibility Principal, proper handling of database connections).
Project and Labs

Each student will select a web application to implement as their *project*. Almost any topic can be selected, but there are some data model restrictions that keep the projects manageable and aligned with the weekly lab assignments. Once a student has completed all the *labs*, they will have created a web application that accesses a database (read, insert, edit, and delete), provides logon/authentication, and enforces authorization rules. Although each student’s web application will employ HTML/CSS to provide some aesthetic appeal, web design is not the focus of the course. We focus on the design and implementation of server code for a reliable, secure, extensible, and maintainable web application. At the end of the semester, the labs (that built up to a single web application) are re-graded as the project. This gives students an opportunity to repair or complete work that may not have been done at the time the labs were due.

Tentative Lab Content – check blackboard for latest lab information and dates.

1. Working with MySqI databases via MySqlWorkbench. Data modeling, SQL.
2. HTML (content), CSS (fonts, layout, colors). Using the JSP include directive for HTML code reuse. Publishing.
3. Creating data display JSP pages. Interfacing JSP code to java classes. Database access from java.
4. Understanding how HTML forms work, how they post data based on the action attribute of the form tag.
5. Handling log on, storing user data into the JSP implicit session object. Security concepts such as how to prevent SQL injection and JavaScript injection hacks.
6. Validating user input. Inserting records into a database.
7. Editing records from a database reusing the same validation code as was employed by insert functionality.
8. Deleting records using a bit of JavaScript to enhance the user interface.
9. Ajax (javascript making asynchronous HTTP requests to Web APIs, writing your own Web API) and Web data formats (XML, JSON)

Tentative Grade Weights

<table>
<thead>
<tr>
<th>Lab Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labs (Homework, 10)</td>
<td>20%</td>
</tr>
<tr>
<td>Lab Challenge</td>
<td>2%</td>
</tr>
<tr>
<td>Lab Quizzes (about 5-6, given in lab, lowest quiz grade dropped)</td>
<td>25%</td>
</tr>
<tr>
<td>Lab Activities (about 5-6, must be completed in lab, lowest grade dropped)</td>
<td>8%</td>
</tr>
<tr>
<td>Project (culmination of all Labs excluding Lab Challenge, code review)</td>
<td>20%</td>
</tr>
<tr>
<td>Comprehensive Test (given last day of class)</td>
<td>25%</td>
</tr>
</tbody>
</table>

Early grade estimates:

- For the Lab Challenge grade (3%), the lab grade is the best estimate.
- For the Project grade (20%), the lab grade is the best estimate.
- So, for early grade estimates, the *lab grade* (actually 22%) is weighted as **50%** (3+20+22) until the other grades become available.
- **Lab activities** are weighted as they will be in the final grade: 5%
- For the Comprehensive Test grade (25%), the quiz grade is the best estimate
- So, for early grade estimates, the *quiz grade* (actually 20%) is weighted as **45%** (25+20) until the comprehensive test grade comes in.
- Also, I do not drop the lowest quiz grade or lab activity grade until later in the semester.
Course Format

- **Weekly Lab Assignments.** Weekly homework will be highly related to (reinforcing) lecture topics.
- **Use of Lab Time.** Every week there will either be a graded lab quiz or lab activity.
  - Short quizzes provide a quick feedback mechanism for you (to see how you are doing) and me (to learn which concepts I need to reinforce). Quizzes are usually closed book, closed computer (paper).
  - Lab activities are short assignments that must be started and completed in lab, then shown to the Lab Assistant for grading.
  - No Make Up Quizzes. If a student misses a lab quiz (or arrives after another student has submitted their quiz and left the room), there is no make-up (for any reason including illness and/or other unavoidable problems). However, every student is allowed to drop their lowest quiz grade.
  - No Make Up Lab Activities. If a student misses a lab activity (or arrives too late to complete the work) for any reason, they get a grade based on what they were able to complete. Every student is allowed to drop their lowest lab activity grade.
  - Exceptions: if a student has a long term documented illness, we might be able to make an accommodation. Ask your instructor if you encounter such a problem.
- **Project.** The culmination of all the labs will be the student's project. While the weekly lab grading may just focus on functional testing, the project will include a code review (done by me) of all the source code, so it's best to keep your code clean and well designed as you go along. The project requires that students have all of their web code functioning and incorporated into a single NetBeans project, with each lab working properly (regression tested).
- **Tutorial.** Each student will write and publish a tutorial on an approved topic. Selected students will present their tutorials to the class towards the end of the semester.
- **Comprehensive Test.** There will be a comprehensive test at the end of the semester (no midterm, no final).

If a student already has extensive background in any topic that is covered, they may substitute other work. These students should discuss their background with the instructor at the beginning of the course.

Labs, Lab Challenge, and Project

- Labs are introduced during lecture. Although there may be some additional instruction at the beginning of lab, the labs are mainly self-directed with help available.
- The most current list of labs (with updated due dates) is listed at the top of blackboard. Please check this single place to learn what lab we are working on and when it is due.
  - Late penalties are based on time of submission of your zip file into blackboard. You are allowed to submit a lab up to 24 hours late with no penalty (as a courtesy, to allow you time to check your coding style and publish before submitting your zip file).
  - If there is ever any question about a lab grade, we go by the code that was submitted into blackboard.
  - If you miss the lab deadline, you can still submit a lab up to 1 week late (also with the 24 hour courtesy) – with a 20% late penalty. Lab homework is not accepted after that, but you still have to complete the work to avoid additional penalties when you submit your project.
- The lab requirements can be found in blackboard and in the 3344 labs page, but DO NOT WORK AHEAD since labs may be changed right up to the lecture when they are assigned. I typically have a "We Are Here" entry in blackboard which I keep moving down as elements are updated for the current semester.
  - If you start working on your lab homework the day before it is due, you risk missing deadlines, doing poorly in the weekly quizzes, and falling farther and farther behind. It is REALLY IMPORTANT that you start early on your labs and ask for help if whenever you are unable to make effective progress.
  - It is also so important to know how to debug, in each of the languages that we use (HTML, CSS, javascript, jQuery, JSP, java, and SQL). I try to include important information about how to debug for each lab.
  - At the end of each lab, there are instructions for lab submission which usually have you publish your code then attach a zip file of all your source code right back into the blackboard assignment (where the instructions were).
• **Lab Challenge.** There are a lot of interesting web development techniques/concepts that are not required to complete your web application project. I have created several "lab challenges" that present some web topic and then ask you to do a simple exercise (that helps you learn and appreciate the concept). Each student will select and complete any one lab challenge. Because lab challenge material is presented in lecture, it may appear on quizzes and the exam.

• **Your project** is your entire web application, the culmination of all your labs (even those labs that you may have missed or not implemented well). Before submitting your project, remove all unused code, perform regression testing, and check that you have met the functional requirements and design specifications of the lab assignments.

• The **CIS department computer labs** are NOT open 24/7. Learn the lab hours and adjust your schedule accordingly. If you want to work from home, *set up your own development environment*. I have tutorials and instructions that can help you with this. Those who promptly set up their own web development environment do much better in this course than those who do not.

**Miscellaneous**

• **Attendance:** If you must miss lecture or lab, check Blackboard to see what material was presented and ask your classmates about anything else that may have been discussed.

• **Communication:** Please contact me as soon as possible if you think you are running into difficulties. Ask me, or your lab instructor, or another student for help AS SOON AS POSSIBLE.

• **Disability Disclosure:** Any student who has a need for accommodation based on the impact of a disability should contact me privately to discuss the specific situation as soon as possible. Student must provide me with a note from the office of Disability Resources and Services (100 Ritter Annex, 215-204-1280).

• **Academic Honesty and Ethics:** Temple University and I expect you to observe the highest ethical standards. When working in the lab or on your project, you may consult others, but the work you submit must be your own. Never share your answers with others. Never accept answers from others. Unless otherwise directed, all quizzes are closed book, closed computer. All violations of academic honesty will be handled according to university policy.

**Grade Scale**

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>93-100</td>
<td>A</td>
</tr>
<tr>
<td>90-92</td>
<td>A-</td>
</tr>
<tr>
<td>87-89</td>
<td>B+</td>
</tr>
<tr>
<td>83-86</td>
<td>B</td>
</tr>
<tr>
<td>73-76</td>
<td>C</td>
</tr>
<tr>
<td>70-72</td>
<td>C-</td>
</tr>
<tr>
<td>67-69</td>
<td>D+</td>
</tr>
<tr>
<td>63-66</td>
<td>D</td>
</tr>
<tr>
<td>60-62</td>
<td>D-</td>
</tr>
<tr>
<td>0-60</td>
<td>F</td>
</tr>
</tbody>
</table>