Office Hours Via Zoom: [https://temple.zoom.us/s/5168459269](https://temple.zoom.us/s/5168459269)
- Wednesdays 11:30 am – 1pm
- Fridays 10 am – 11:30 am
- And by appointment (email me at sallyk@temple.edu to set up a time).
- NOTE: I have found that using Zoom screen sharing is a wonderfully effective way for me to assist students with programming/technical problems.

FALL 2020 – This course is being offered online.
- You’ll need a good computer (to install your IDE), a reasonable internet connection, and a web cam and (for taking tests). I’d suggest a PC/MAC with at least 8 gigs of RAM.
- Be available during the times that the course is scheduled - for synchronous activities like tests and group learning. However, I may substitute an asynchronous activity for the scheduled class time.
- If you have a DRS accommodation for extended time to take tests, make sure you have the extra time that you need – either before or after your scheduled lecture time.

Course Description

In this course students will learn how to design and implement web applications – both server side and client side code. We use open source (free) development tools such as the Netbeans Bundle (includes NetBeans sensitive editor / compiler as well as Apache Web Server and Glassfish JSP application server), MySQL (Database Management System) and MySQL Workbench (GUI to help you create SQL to interact with a database). In this course, students write code in the following languages:
- HTML (structure and content of web page),
- CSS (styling of web page),
- JavaScript (web page functionality)
- Java/JSP (sever side code that we use to create Web APIs that provide database access using SQL within Java).
- SQL (to create and modify data).
In this course, students shall:

- Create a web application UI (User Interface) using HTML and CSS. Use JavaScript for client side functionality.
- Create their own MySql database and populate it with data.
- Publish their web applications using the sftp protocol.
- Test their web applications using the https protocol.
- Create Web APIs (server side Java/JSP code that access their database) that output produce data in the JSON web format which is easily converted to JavaScript objects (client side) and to Java objects (server side).
- Write client side code (JavaScript/AJAX) to invoke the Web APIs.
- Use server side objects such as out, request, and session (out is how you write to the page, request contains URL input parameters, session is what we use for log in).
- Implement log on and learn about web application security.
- Apply software design patterns such as MVC (Model-View-Controller), SRP (Single Responsibility Principle), DRY (Don’t Repeat Yourself), and Dependency Injection (if consumer code calls provider code, provider code shall not access any aspect of consumer code – except for whatever consumer code passed to provider code).

Although each student’s web application will employ HTML/CSS to provide some aesthetic appeal, web design is not the major thrust of this course. Instead, this course focuses on designing and implementing client side and server side code to create a reliable, secure, extensible, and maintainable web application. We will focus on the enduring basics of web development (HTML, CSS, JavaScript, Server Side Programming), but we will also learn a bit about ReAct which is (at this moment) a popular JavaScript Single Page Application Framework. Students must recognize that frameworks come and go but the basics are enduring.

Course 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 students have knowledge and skills obtained in the pre-requisite courses listed just 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 assignments (if they wish), as long as they meet all homework requirements and submit on time.

Textbook

There is no text book. Web references and other materials will be posted online.
Course Format

- **Homeworks.** Almost every week, there will be a programming assignment that is highly related to (reinforcing) lecture topics. To get a grade for your homework you must:
  - Complete the homework assignment and test it locally. Then, publish it and test what you published.
  - Upload a zip file of your WHOLE web application (NetBeans project) into Canvas.
  - If you have not completed your homework by the due date, you can still complete it by the following week (with a -20% penalty). After that, homeworks are not accepted, but you still have to complete the work to avoid further deductions on your project grade.
  - If there is any question about a Homework grade, we go by the code that you uploaded into Canvas.

- **Project.** Your project is the culmination of all your labs (all combined into a SINGLE, regression tested, web application). Your project grade is based on functional testing plus a code review of all your source code. So, each week, be sure to keep your code well designed/organized, bug free, using self documenting names and adequate comments. EVERY WEEK, your NetBeans Project shall grow – don’t create a new project each week.

- **Lab Activities.** Almost every week, there will be a graded lab activity.
  - Lab activities are short exercises typically designed to give you the skills you need to get started on your homework. Normally, a Lab Activity asks you to do a subset of the homework assignment.
  - At the end of the lab activity, students upload a zip file of their project into Canvas and also publish their code (to their Lab Activity Web Site). There is no late submission policy for lab activities. If you don’t meet the deadline, you don’t get a grade for the lab activity. (However, if a student has a long term documented illness, let the instructor know – perhaps an accommodation can be reached.)
  - To account for unexpected problems, your lowest Lab Activity grade will be dropped.
  - Each student will have two web folders into which they publish – one for publishing Lab Activities and one for publishing homework (which also “grows into” your project).
  - Any questions about Lab Activity grade will be based on the code submitted into Canvas.

- **Quizzes and Tests.** The tentative plan is for 1 quiz and 3 tests (and no final exam). **The only way to do well on quizzes/tests is to experiment a lot with sample code and homework code.**

- **Tutorial Option.** This course has a lot of sample code so we are able to cover all the material that is needed for even beginning web development. However, many students prefer to write code from scratch (which I totally understand). You are in luck – select the Tutorial option where you can can pick from a wide array of topics. Those who select the Tutorial Option do not have to do the Insert Homework.
  - In order to get a course grade better than B you must select the Tutorial Option (and get a B or better on it). This is a challenging assignment that demonstrates mastery of the course material.
  - If you select the Tutorial Option, you must submit a Tutorial Proposal (worth 0 points) by its due date.
  - If you are struggling with homeworks, it may be best not to select the tutorial option because it is a very open ended assignment with no sample code.
  - The tutorial involves writing extensive JS code (consumer/provider style), explaining the code, and presenting to the class.
Syllabus for CIS 3308 Web Application Programming

- **Synchronous Class Activities.** Although I have a lot of materials for you to read and videos for you to watch (at your own convenience), there may be other activities that we do together as a class using Zoom. By clicking on the Syllabus link in Canvas, you should be able to see a complete list of all class expectations and due dates – including scheduled synchronous class activities (that may or may not have a grade associated with them), but your participation will be noted.

- **Weekly Routine.** The weekly routine for this course is typically:
  
  - Listen to a video that introduces the module and the homework for the module.
  - Do a Lab Activity (that gets you started on the homework for the module) – this is due by the end of your Lab Period.
  - Study sample code and other lecture materials, e.g., videos that explain the code and related concepts while working on your homework for the week.
  - Publish and submit your homework before the start of your next lab period.

**Tentative (and approximate) Grade Weights**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab Activities, must be completed in lab (about 11, drop lowest)</td>
<td>10%</td>
</tr>
<tr>
<td>Participation in Synchronous Class Activities (To Be Decided)</td>
<td>5%</td>
</tr>
<tr>
<td>Programming Homeworks (about 9)</td>
<td></td>
</tr>
<tr>
<td>OR if Tutorial selected (about 8 HWs plus Tutorial)</td>
<td>20%</td>
</tr>
<tr>
<td>Project (culmination of all Homeworks/Tutorial plus code review)</td>
<td>15%</td>
</tr>
<tr>
<td>1 Quiz</td>
<td>5%</td>
</tr>
<tr>
<td>3 Tests (no final exam)</td>
<td>45%</td>
</tr>
</tbody>
</table>

**100%**

**EXCEPTIONS:**

- If the average of your test grades is less than C-, your course grade will also be less than C- (and you need at least C- to be able to count this course towards CS major requirements).
- To get a course grade higher than B, you must do the Tutorial (and get a B or better on it).

**Temple Grade Scale**

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>93-100</td>
</tr>
<tr>
<td>A-</td>
<td>90-92</td>
</tr>
<tr>
<td>B</td>
<td>83-86</td>
</tr>
<tr>
<td>B-</td>
<td>80-82</td>
</tr>
<tr>
<td>B+</td>
<td>77-79</td>
</tr>
<tr>
<td>C</td>
<td>73-76</td>
</tr>
<tr>
<td>C-</td>
<td>70-72</td>
</tr>
<tr>
<td>C+</td>
<td>67-69</td>
</tr>
<tr>
<td>D</td>
<td>63-66</td>
</tr>
<tr>
<td>D-</td>
<td>60-62</td>
</tr>
<tr>
<td>D+</td>
<td>0-60</td>
</tr>
</tbody>
</table>

4
Syllabus for CIS 3308 Web Application Programming

Tentative Course Schedule - To see actual dates click on “Syllabus” from your Canvas Course.

Each homework usually takes one week to complete (except as listed below).

- Homework 1: Home Page
- Homework 2: Database Set Up and JavaScript Routing
  - Quiz on HTML, CSS, and introductory JS
- Homework 3: Slide Show (JavaScript, AJAX, JSON)
- Homework 4: Data Display (more practice with JS, AJAX, and JSON)
  - Test on database concepts and advanced JS, AJAX, and JSON
  - Tutorial Proposal due for students who will pick the Tutorial Option
- Homework 5: Web APIs (the first homework that uses server side programming).
- Homework 6: Log On (2 weeks)
  - Test on Java/JSP DB access code, Web App Security, using JS to create interfaces for user input.
- Homework 7: Insert Homework or Tutorial Option (2 weeks)
  - Lab Activity on React, a JavaScript Single Page Application Framework.
- Homework 8: Update (2 weeks)
  - Student Tutorial Presentations
- Homework 9: Delete
  - Comprehensive Test focusing on using JS to create interfaces for user input and validation messages, more on server side database access programming and web security.

Coverage of Concepts in the Homework Assignments

<table>
<thead>
<tr>
<th></th>
<th>HTML CSS</th>
<th>JS</th>
<th>Database</th>
<th>AJAX JSON</th>
<th>JSPs/Web APIs tunneling in enhanced instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Homepage</td>
<td>√√</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Database &amp; Routing</td>
<td>√</td>
<td>√</td>
<td>√√</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Slideshow</td>
<td>√</td>
<td>√√</td>
<td></td>
<td>√√</td>
</tr>
<tr>
<td>4</td>
<td>Display Data</td>
<td>√</td>
<td>√√</td>
<td></td>
<td>√√</td>
</tr>
<tr>
<td>5</td>
<td>Web APIs</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>6</td>
<td>Log On</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>7</td>
<td>Insert</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>8</td>
<td>Update</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>9</td>
<td>Delete</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>
Syllabus for CIS 3308 Web Application Programming

Miscellaneous

- **Set up your development environment on your PC/MAC ASAP** so you can work from home. The CIS 3308 Web Page has instructions to help you with this (google “SallyK Temple”, click on Teaching – CIS 3308).
- **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.
- **Attendance:** All “lectures” will be recorded and available for students to watch at their convenience but students need to be available during scheduled class times for synchronous activities like tests. **If you have a DRS accommodation, make sure you have enough time either before or after your scheduled lecture time.**
- **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:** You are expected 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. All violations of academic honesty will be handled according to university policy.

**Fall 2020 IMPORTANT DATES:**

- Monday 8/24: Start of semester
- Tuesday 9/8: Last Day to Drop (you don’t pay for the course and it will not appear on your transcript).
- Monday 3/12: Mid Term Ratings end.
- Tuesday 10/20: Last Day to Withdraw (you pay for the course, W stays on your transcript, no effect to GPA).
- Monday 11/23 – Friday 11/27: Fall Break (includes Thanksgiving)
- Monday 12/7: Last day of classes