CSCI 237: Computer Organization
Syllabus for Spring 2018

General Info

Instructor: Jeannie Albrecht
Email: jeannie@cs.williams.edu
Office: TCL 304
Lectures: MWF 12:00-12:50 in Schow 030A
Labs: R 1-2:30 and 2:30-4 in TCL 217 and TPL 312
Lab Instructor: Steve Van Wert
Textbooks: Computer Systems: A Programmer’s Perspective (3rd Edition),
by Randal E. Bryant and David R. O’Hallaron (required)
The C Programming Language (2nd Edition),
by Brian W. Kernighan and Dennis M. Ritchie (recommended)

Course Description

This course provides a programmer’s view of how computer systems execute programs, store information, and communicate. It enables students to become more effective programmers, especially in dealing with issues of performance, portability and robustness. It also serves as a foundation for courses on compilers, networks, operating systems, distributed systems, and graphics, where a deeper understanding of systems-level issues is required. At the same time, a model of computer hardware organization is developed from the gate level upward. Topics covered include: machine-level code and its generation, performance evaluation and optimization, computer arithmetic, memory organization and management, and (maybe) networking protocols and supporting concurrent computation.

Grading Details

Grades will be computed as follows:
5% Class Participation
45% Programming Labs
25% Midterm Exam
25% Final Exam

Each of these items are explained in detail in the following sections.

Class Participation

Lectures are mandatory and you are expected to attend regularly. One goal of this course is to promote discussion of the assigned topics among all class members. As such, you are encouraged to ask questions, point out problems, and make observations during class.
Programming Labs

There are weekly programming lab assignments. All lab programs are graded on design, documentation and style, correctness, and efficiency. Programs should be turned in electronically by the posted due date. Each student may use a maximum of **three free late days** during the course of the semester. A late day permits you to hand in an assignment up to 24 hours late, without penalty. You may not use more than two on any single assignment. Once those late days are exhausted, late labs will be penalized 20% per day. Labs will not be accepted more than four days late. To use a late day, you must email your instructor in advance and include a note to the graders in the comments at the top of your files. **Attendance in lab is mandatory. Repeated absence or tardiness to lab will result in failure of the course.**

Note that although some of us will meet in TCL 217 (a Mac lab), we will primarily use the Computer Science Department's Linux computers for our programming assignments. Most assignments will be completed using the C programming language. More complex labs will be worth more points towards your final grade than simpler labs.

Midterm and Final Exam

There is a written midterm and final examination in this course. The exams are closed book, closed notes, and stress conceptual understanding of the material. Details regarding the specific format of the exams will be discussed in class.

Collaboration

Unless specified, all programming assignments and examinations are to be completed individually. I encourage collaboration and assistance in understanding material, but not in developing solutions. **Please be sure to give explicit credit for any help received.** If you have any doubts about this, ask me whether or not collaboration is appropriate. Uncredited collaborations will be considered a violation of the honor code. The Computer Science honor code and computer usage policy applies to all material in this class. Please review this info (link available on course webpage) if necessary.

Calendar

The following calendar is a (very) tentative schedule of topics that we will cover in class.

<table>
<thead>
<tr>
<th>Week 1</th>
<th>Representing and Manipulating Information (CSAPP Ch 1–2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weeks 2–3</td>
<td>Machine-Level Representation of Programs (CSAPP Ch 3)</td>
</tr>
<tr>
<td>Weeks 4–6</td>
<td>Processor Architecture (CSAPP Ch 4)</td>
</tr>
<tr>
<td>Week 7</td>
<td>Optimizing Program Performance (CSAPP Ch 5)</td>
</tr>
<tr>
<td>Week 8</td>
<td>The Memory Hierarchy (CSAPP Ch 6)</td>
</tr>
<tr>
<td>Weeks 9–10</td>
<td>Virtual Memory (CSAPP Ch 9)</td>
</tr>
<tr>
<td>Week 11</td>
<td>Network Programming (CSAPP Ch 11)</td>
</tr>
<tr>
<td>Week 12</td>
<td>Concurrent Programming (CSAPP Ch 12)</td>
</tr>
</tbody>
</table>