Karen Miller

http://pages.cs.wisc.edu/cs354-1

smoler@cs.wisc.edu
office phone: 608-263-1724
4283 CS
Karen's background

- BS EECS at Berkeley
- Worked at Astronautics Corp. of America doing HW design the memory system of the ZS 1 (mini supercomputer)
- MS CS here at UW Madison
- Many semesters teaching
- Many years with the HTCondor project
Course Goals

- learn about **computer architecture**
- see more of *the big picture* of how computers work, in both the HW and SW
- become a **C programmer**
- be introduced to many aspects of **operating systems**
Course Grades

- Standard UW grades: A, AB, B, BC, etc.
- 90% of available points will be an A
- Course grade based on
  - 30% assignments
  - 20% exam 1
  - 25% exam 2
  - 25% exam 3 (Friday May 15, 7:45 am)
Schedule (from the course web page)

Dates

Topics - the subject and links to lecture slides

Readings - from the textbook, written by Karen, from K&R, outside materials

On Your Own - you are responsible for learning these topics, but they will not be covered in lecture

Beyond 354 - material of interest, but not part of the course

Due Dates
Lecture Slides (always provided)

- reduce note-taking time
- are *incomplete*
- do **not** substitute for attending lecture
Assignments

5 graded assignments
● most are programs
● work alone or in a pair
● grading is *picky*

Karen's advice:

*start assignments early*
Computer Systems: A Programmer's Perspective, by Bryant and O'Hallaron, second edition

And, obtain a reference manual for programming in C. Online references might be enough. Preferred reference:

The C Programming Language by Kernighan and Ritchie (K & R), second edition
Use all your resources:

Come to class (you are responsible for all class material). Know the prerequisite material. Learn the On Your Own subjects. Form a study group. Do the assignments! Use the lecture slides. Do the readings. Ask questions in office hours (both Karen's and the TAs').
Course Topics

- A common HLL: C
- x86 assembly language
- arrays, stacks, and queues
- implementing functions
- I/O, including exception handling (interrupts and traps)
- memory hierarchies and caching
- virtual memory
- linking and loading
- networks and network programming (maybe)