# CSE 471: Computer Design and Organization
# Spring 2009
# Lectures: TTh 10:30-11:50 AC 503
# Section: Th 2:30-3:20 TBD

### Instructor

Susan Eggers, CSE 586, 543-2118, eggers@cs.washington.edu
Office Hours: Tuesdays, 1:30 - 2:20

### TA

Vince Zanella, CSE 216, vincezanella@yahoo.com
Office Hours: Wednesdays, 2:30 – 3:20 and Fridays, 12:00 – 12:50

### Course Material

The purpose of this course is to give you a broad understanding of the concepts behind several advanced microarchitectural features in today's microprocessors and to illustrate those concepts with appropriate (usually modern) machine examples. We will cover the rationale for and the designs of strategies for dynamic branch prediction, multiple-instruction issue, dynamic (out-of-order) instruction scheduling, (maybe) caching techniques, chip multiprocessors, multithreaded processors, and dataflow machines. Some of these topics depend on material you covered in 378; for these, we'll quickly review that material, and then go on from there.

You'll augment your knowledge of the architectural concepts and designs by doing experimental studies that examine and compare the performance of several alternative implementations for a particular feature. Here you'll learn: (1) how to pose an architectural research problem, (2) how to determine the severity of the problem in practice, (3) how to design architectural experiments, (2) how to choose metrics that best illustrate a feature's performance, (4) how to analyze performance data and (5) how to write up your experiment and results – all skills you'll need if you plan to do computer evaluation either in development or research, and in any applied subfield of computer science, not just computer architecture.

You must have already taken 370 and 378 in order to take this class.

### Reading

Most reading assignments will be taken from *Computer Architecture: A Quantitative Approach* by John L. Hennessy & David A. Patterson, Morgan Kaufmann, 4th edition, 2007. To get the most out of the lectures, **read the material *before* topics are discussed in class**. There will also be some supplementary reading that you will be able to access from the course web pages. And the project will involve reading research papers.

### Schedule

There is a daily schedule in the course web area. The schedule will tell you what topics we will cover and when, what reading should be done before you come to a particular lecture, and when projects are due and exams will be held. I'll be updating this schedule continuously, as I plan each set of lectures. So you should check it frequently, so that you can anticipate what material we'll be covering.

### Class Discussion

Before each class, think about what you've read for the upcoming lecture and about the material in the previous lecture and come prepared to pose and answer questions, present your views of the architectural schemes we discuss and offer alternatives. Although I'll be lecturing, this class is part seminar. And the more it's seminar-like, the more fun we'll have!

### Exams

There will be two midterms, one in the middle of the quarter and one on the regular 471 exam date. The one at the end will cover material from the second part of the course only.

### Projects

There will be two kinds of projects, one short and one long. The short one is pretty cut and dry. It's an experimental study that will give you experience in designing architectural experiments and evaluating architecture features, and will hone your intuitions about the performance ramifications of changing certain aspects of their implementation. You'll also make your first stab at writing up your results in research-paper style. With that experience under your belts, the second project will involve research in one of the hot topics in computer architecture at the moment, multi-core processors, and will span most of the quarter. Here you'll be investigating issues no one has investigated yet and designing new architecture schemes and/or architecture-related compiler algorithms to solve what are still outstanding problems. This will of course involve creating or adding to the infrastructure to do the experiments. It will be very important to continually make steady progress on this, as it is impossible to finish a research project by pulling a series of all-nighters at the end of the quarter.

In the discussion sections Vince will explain more about the two projects and how to use the simulators. We'll use the SimpleScalar uniprocessor simulator for the first project and software that we've not yet chosen for the large project.

You're welcome to work in teams of two students for each project. Try to work with a different partner for each.

All project reports are due at the beginning of class; no late assignments will be accepted.

### Machines

We'll be using attu, a cluster of four Dual Pentium IV 2.8GHz Xeon servers with 4GB of memory and running Linux 2.6.12. You may use the workstations in AC 002, AC 006 or AC 022 to log into attu. Alternatively you may use other computers to log in.

### Grading

Grades will be computed using the following **approximate** weighting: the first midterm = 20%, the second midterm = 30%, and the projects = 50% for all. This may change somewhat, depending on the size of the projects.

### Collaboration

Discussing the course content with fellow students is an effective way to learn the material, and is encouraged. However, exams must represent your own mastery of the material, and projects must represent the contribution of your team.

### Communicating

We will communicate a lot through e-mail. Vince and I will be emailing out assignments and clarifications of the assignments, if needed. And you should use e-mail for asking and answering each others' questions. (But if you have questions that need a detailed or long explanation, it would be much easier to come to our office hours.) Therefore you should register on the class mailing list **immediately**. To add yourself to the class email list, follow the instructions on the 471 web pages.