

# CS 4210 Advanced Operating Systems and CS8803 Graduate Introduction to Operating Systems Fall 2012 Syllabus

**Meeting Times:** Tue-Thur, 1:35-2:55PM, CCB 52

**Instructor:** Ada Gavrilovska,  
KACB 3228, 404 894 0387  
ada@cc.gatech.edu

**Office Hours:** TBD

**Teaching Assistant:** Jai Dayal, dayalsoap@gmail.com

**TA Office Hours:** TBD

## Course Outline:

We will discuss operating system abstractions and their implementations. The core of the course contains concurrent programming (threads and synchronization), inter address communication, and an introduction to distributed operating systems. Other topics may be added, especially in conjunction with related programming projects.

## Textbooks:

No textbook is strictly required, however Silberschatz et al. is recommended. In addition, I will be passing out handouts, and make research papers available online.

- **Operating Systems Concepts**, Silberschatz and Galvin, 7<sup>th</sup> or 8<sup>th</sup> Ed. Addison-Wesley. (earlier editions should be fine)

Here are some suggested background books:

- **Distributed Systems**, Tanenbaum and VanSteen. Prentice Hall.
- **Modern Operating Systems**, A. Tanenbaum, 3<sup>rd</sup> edition, Prentice Hall, 2007.
- **Multithreaded Programming with Pthreads**, Lewis and Berg. Prentice Hall. (Excellent book on multithreading and systems issues—not too Pthread-specific at all.)
- **Pthreads Programming**, Nichols, Buttlar, Farrell. O'Reilly. (Fairly good and inexpensive Pthreads manual.)

## Exams:

All exams are closed-books, closed-notes. Current *tentative* schedule:

Midterm Exam: TBA, in class

Final Exam: Tue. Dec. 11<sup>th</sup>, 2 2:50-5:40pm (based on Georgia Tech official Final Exam Calendar)

## Grading:

55% exams: 25%, 30%

45% programming assignments

P/F students must have passing grade in both components to pass.

## Homework and Assignment Due Times:

Programming assignments are due by midnight (11:59pm) of the due date. All submissions are handled through T-square.

**Late Penalty Policy (only for programming assignments):**

This late penalty policy applies **only to programming assignments/projects**. 5% penalty for each day late, up to 5 days (25%).

**Honor Code:**

The Institute Student Conduct Code is printed on pages 336-339 of the GT General Catalog. You and I are expected to abide by it. I do not deal with cases of academic misconduct in person (the Office of the Senior Associate Dean of Students will be notified).

**Other Resources:**

*T-square*: Make sure you have access to the CS-4210-A, CS-8803-GOS t-square site. The site includes Wiki pages with Lecture Schedule, Resources pages with all handouts and papers, etc. All announcements, assignments, and email communication will be handled through this site.

**Other Issues:****Test make-ups:**

You miss an exam, you lose the points! If you have a serious reason to miss a scheduled exam (e.g., sickness, or serious reason to travel) make sure you tell me *in advance* of the exam. After the exam is over, if you have not taken it and you have not told me why you didn't take it, you are out of luck! Emergencies are an exception, of course, but "I was stuck in the dentist's office – he said he'd see me at 1 but didn't until 3" is not an emergency! Also, *interviews are not an excuse for a make-up test* – the exam dates will be provided by the end of the first week of semester, so please work with the recruiters to arrange for non-conflicting dates.

**Collaboration:**

No collaboration is allowed across groups for projects. Specifically, showing someone your source code is not allowed, unless you only show small fragments (say, 10 reasonably formatted lines) and these only concern low-level technical issues. Collaboration on how to solve specific technical problems is allowed (and encouraged) through the t-square chat room or forums. The rule-of-thumb for collaboration is: if you feel comfortable posting the info to the t-square site (for everyone, including me, to see), then it is ok. Otherwise, it is not.

**Debugging with me:**

We (TA and instructor) will spend time helping you debug your programs, but this typically will happen only when you are totally lost. If you come to us with a problem, make sure you have spent a lot of time trying to resolve it. Also, make sure you have a setup where we can quickly reproduce the problem, and ideally even a much simplified example where the problem still exhibits itself.

**Project descriptions:**

For *any* major departure from the project requirements (different deliverable, very different mechanism, etc.) consult with me and the TA first! Do not assume that the project is just about testing whether you can program! It is not! The project is about implementing what we ask for. If we say "use RPC" and you instead decide to use sockets, you will lose many points, even if you are an expert programmer, and you have a great deliverable, and you spent twice the time that everybody else did coding your project to perfection.