CS350
Fundamentals of Computing Systems
Spring 2009

General Information:

- Instructor: Rich West
  - Email: richwest@cs.bu.edu
  - Office hours: Thu 5:00-7:00pm, Fri 12:00-1:00pm
  - Office: MCS 289
- TF: Likai Lui
  - Email: liulk@cs.bu.edu
  - Office hours: TBA
- Sections: Fri 11:00am-12:00pm, 1:00pm-2:00pm, B31
  - Students should attend one of the above sessions each week.
- Class location: CAS 313
- Lecture times: TR 3:30-5:00pm

- Course mailing list: cascs350a1-l@bu.edu
  - NB: All registered students are automatically added to the course mailing list.
  - Use the above mailing address to submit general comments and questions to the class.

NOTE: The last day to drop a course without a 'W' grade is TBA. The Registrar's website provides further information about important dates.

Overview:
This course is a required sophomore/junior-level Computer Science course. It covers the fundamental concepts underlying the design and implementation of computing systems.

The philosophy underlying the design of this course is that students should be familiarized with problems that reoccur in software systems, and should be acquainted with the set of classical algorithms/techniques for solving such problems. In particular, it is important to develop the ability to recognize standard problems in different wordings and within unusual context, and match them with appropriate solutions.

Catalog Description:

Rigorous treatment of invariant concepts, algorithms, and performance evaluation methods underlying computing systems design. Topics include modeling and analysis of concurrent processing, computational resource scheduling and consumption, and performance evaluation techniques.

Prerequisite Courses:
The following courses are required.
CS-210: Basic computer organization and software/hardware interface concepts, which are covered in CS-210.
CS-112: Programming and basic algorithmic thinking.
CS-237: Elements of combinatorics and of discrete and continuous probabilistic analysis (also covered in MA-293).

Recommended Courses:
The following courses, while not required, are recommended for a better assimilation of some of the topics covered in the class.
CS-235: Concepts related to graphs and graph properties (also covered in MA-294).
MA-381: Elements of probability and statistics, including basic understanding of probability distributions, expectations, and correlation metrics.

Recommended Follow-up Courses:
The following courses build on knowledge assimilated in CS-350. They are natural follow-up courses in decreasing order of relevance.
CS-552: Introduction to Operating Systems (and if you like that, then try out CS-553 for some hand-on experience with OSes)
CS-455: Introduction to Networking (and if you like that, then try out CS-556 or CS-559 for advanced networking)
CS-410: Software Systems and Systems programming
CS-470: Performance Evaluation and Modeling of Computing Systems and Networks
CS-450: Introduction to Computer Architecture (and if you like that, then try out CS-550 for advanced/quantitative Computer Architecture)
CS-560: Introduction to Database and Information Systems (and if you like that, then try out CS-562 or CS-565 for advanced DB courses)

Function in the Curriculum:
As a core undergraduate course, CS-350 does not focus on particular implementations or specific technologies. On the contrary, it stresses the fundamental concepts and basic algorithms that have survived (and are likely to survive) the evolution of computer software systems in general, and operating systems in particular.

This course is significantly different in purpose and coverage from CS-210 (Computer Systems) and CS-410 (Software Systems). CS-210 and CS-410 introduce computer systems to sophomores using a hands-on approach by examining the "mechanics" and/or the "plumbing" of modern (or typical) computer operation through exposure to various interfaces between architecture, compilers, loaders, linkers, and run-time systems. CS-350 does not focus on a particular interface or a particular technology, rather it deals with fundamental notions and algorithms that are common to computing systems

in general and to software systems in particular.

This course is required for the following courses: CS-552 (Operating Systems), CS-550 (Advanced Computer Architecture), CS-551 (Parallel Computing Architectures and Models), CS-455/655 (Data Communications), CS-560 (Databases), and CS 470/670 (Performance Analysis). In that respect, this course covers basic notions that we expect students (graduates and undergraduates alike) enrolling into these classes to have mastered. For example, it is inconceivable that a student who is interested in taking a database or an operating systems class, not to have been exposed to the notion of concurrent processing and mutual exclusion. Similarly, it is inconceivable for a student enrolled in a networking, database, or operating systems class, not to have been exposed to performance analysis using simple queuing systems, for example.

Topical contents:
The topics covered in this class could be grouped under 3 general themes:

- Performance Analysis and Evaluation (including a crash introduction to probability)
- Scheduling and Resource Management
- Concurrency and Synchronization

It is very important to realize that this is NOT an Operating Systems course. Despite the nature of the textbook, this course will not get into specifics of operating systems (e.g. API, Unix/Solaris/NT implementations). Of course, the concepts studied in this course are central to more advanced system courses (including Operating Systems, Networking, and Databases, as discussed above).

Course Lectures:
Class participation and course attendance will account for 5% of the final class grade. Questions are eoncouraged during lecture time. Significantly, asking questions is an essential aspect of education so do not be afraid by what you may think is a "stupid question". If you have reason to ask, chances are there will be others in the class who will be glad you asked the question.

Course Discussion Sections:
If you are taking this class then you need to sign up for a weekly one-hour discussion session. Please check the course web page for the coordinates of your teaching fellow and of the place/time of the discussion sections.

The teaching fellow will be leading the discussion sessions. Materials covered in the discussion sessions will either be elaborations on topics covered in the lectures, or complementary information to help reinforce the concepts being taught. Discussion sessions will be the venue used to answer any questions (or provide clarifications) regarding the homework assignments.

If (for some reason) you miss (or cannot make) the discussion section for which you are signed up, then please make sure to attend another one in the same week.

Office Hours:
You should come to the Instructor and/or Teaching Fellow's office hours with your questions well prepared.

Office hours are NOT meant to be tutoring sessions; they are meant to answer specific questions about the material covered in lectures, discussion sections, and textbook(s). Questions like "could you repeat your explanation of ..." or "I do not understand section ... of the textbook" should be asked in the lecture and/or discussion sections.

Mailing List:
There is a course mailing list. Further information will be made available in class. You should check email regularly, for updates and information concerning the course. You are free to use the mailing list to ask general questions, or to share useful information with the class. However, the class mailing list should not be abused. Typical rules applying to mailing list and newsgroup etiquette apply. Do not use the mailing list to disseminate spam, abusive remarks and so forth. Similarly, discretion is advisable when deciding to post a message, especially if you feel it may reveal answers to homework questions, or impact other class participants.

Grading:
The minimum grade for this course to count towards the CS concentration is C.

The instructor is not allowed to give W (withdrawal) grades. One can receive such a grade only by dropping this class by the deadline specified by the registrar office for withdrawals with or without a W grade (check the registrar's office calendar for the exact date). Also, an incomplete "I" grade cannot be given, except if a student misses completing assignments and/or misses taking tests due to circumstances beyond their control.

Grade Breakdown:
Grading is, tentatively, broken down as follows:

Homework Assignments	35%
Midterm Exam(s)	25%
Final Exam	35%
Class Participation	5%
Total:	**100%(+ bonuses)**

- Grading is subject to change.
- Possible bonuses may be awarded for assignment extensions.

Homework Assignments:

Homework assignments constitute an important part of this course. They are designed to

help you understand the materials covered in lectures and in assigned readings. It is only by doing the homework that you really learn the material. Homework assignments will NOT be weighted equally and problems within homework assignments may also vary in their relative weights. Harder/longer homework assignments will constitute a larger percent of the combined grade for all homework assignments.

Some of the problems in the homework assignments may require programming. Students will be expected to tackle problems that expose them to challenges centered around topics such as performance evaluation, modeling and simulation, and concurrent programming.

Where appropriate, homeworks are to be submitted via the slotted homework box, labeled ``CS-350 / Drop Box'', in the hall outside of MCS 137. All homeworks are due before the start of lecture on the deadline date, unless otherwise stated.

A late homework must be time-stamped by a CS Office staff and left in the Teaching Fellow's mailbox (i.e. NEITHER in the Drop Box NOR in my Mailbox). Do not hand in your homework in class or during office hours. Do not hand in your homework by slipping them under the office door of the instructor.

There will be a penalty of 25% for a homework handed in one class late, and of 50% for a homework handed in one week late. You may hand in part of the homework by the deadline to avoid the penalty on that part. No homework will be accepted if late by more than one week. There will be NO exceptions to this policy, other than for religious holidays and certified medical excuses. In such cases, extensions will be granted only if (and until) the homework solutions are posted (hopefully, about 1 week after the original due date).

Interim Exams:
There will be at least one midterm exam. The date(s) and information concerning all in-class exams will be announced at least one week before they are to take place.

Final Exam:
Information about the date, time and location of the final exam will be announced when known. As with interim exams, details about the final will be provided by the instructor before the end of the semester.

Policy on Missed Exams:
Please mark the exam dates on your calendar (and remember them when you make your recess and end-of-semester travel plans!) There will be ABSOLUTELY NO make-up exams, except for medical emergencies and religious observances. For medical emergencies, you must provide a letter from a doctor, specifying the period of time during which you were unable to attend an exam. Anticipated absence from exams due to religious observances must be identified at the beginning of the semester.

Academic Code of Conduct:
In addition to the normal and well-understood strictures against cheating on exams,

altering transcripts and so forth, there are other varieties of academic misconduct described in the Academic Conduct Code of Boston University which you must be aware of when working on assignments. The most relevant sections (B, F, and I) of the Academic Code for the assignments in this class are as follows:

> *Plagiarism. Any attempt by a student to represent the work of another as his or her own. This includes:*
>
> - *Copying the answer of another student on an examination or copying or substantially restating the work of another person or persons in any oral or written work without citing the appropriate source, and collaborating with someone else in an academic endeavor without acknowledging his or her contribution.*
> - *Knowingly allowing another student to represent your work as his or her own.*
> - *Submitting the same work in more than one course without the consent of the instructors.*

Policy on Collaboration:
There is nothing wrong in principle about discussing the topics covered in the course with your friends and colleagues. However, there are severe consequences to *plagiarism*. In particular, when you submit a homework with your name on it, you are claiming that the work contained therein is your own; it is *Plagiarism* to submit work under your own name in which:

- You collaborated with another student (current, former, friend, etc.) in solving the homework problems;
- You copied the solution from another student; or
- You obtained the solution from a book or other sources.

The above is nothing less than *plagiarism* and will be punished accordingly.

If you are in a situation whereby you have collaborated with a student on a homework problem, or obtained homework solutions (or parts thereof) from some other source, then you should CLEARLY and UNAMBIGUOUSLY disclose this in your write-up. *Failure to do so constitutes plagiarism*.

If you are caught cheating in a test or plagiarizing a homework, you will AUTOMATICALLY receive an F and the matter will be reported to CAS Academic Conduct Committee, which recommends sanctions to the Dean of the College. Handing in your own assignment a day or two late will affect your grade FAR LESS than turning in a copy of someone else's work on time!

For more information on the Academic Code of Conduct for the College, please refer to the web page available at http://www.bu.edu/cas/academics/programs/conductcode.html

---

Class Notes:

- Azer Bestavros, Lecture Notes on Fundamentals of Computing Systems.
    - We will refer to Professor A. Bestavros' lecture notes, prepared as part of earlier instances of this course, where appropriate.

References (optional):

- Silberschatz, Galvin and Gagne, "Operating System Concepts," John Wiley and Sons, Inc., 2004 or more recent edition - ISBN: 0-471-69466-5.
- William Stallings, Operating Systems (Internals and Design Principles), 5/e or 4/e. Published 2004/2002 by Prentice Hall 779 pp. ISBN 0-13-031999-6 (4/e) or 0-13-147954-7 (5/e).
- Arnold Allen, Probability, Statistics, and Queueing Theory with Computer Science Applications, 2/e, Academic Press, hardcover, 740 pages, 1990., ISBN 0-12-051051-0.

Software and Programming Information:

Primarily this course is intended to introduce the underlying concepts relevant to computing systems. To reinforce these concepts we will, where appropriate, implement programmatic solutions to various problems. These problems may involve concurrency and synchronization. Additional software information will be provided as necessary, to help with any programming assignments in the course.

Tentative Course Schedule/Topic List

	Topic
	A Bird's Eye View of Systems and System Abstractions
	Performance Metrics of Computing Systems
	Elementary Probability Analysis
	Probability Distributions and Expectations
	Elementary Queuing Analysis: The M/M/1 Queuing Model
	Empirical Performance Evaluation: Discrete Event Simulation
	Elementary Statistics: Sampling, Estimation, and Confidence Intervals
	Elementary Queuing Analysis: Variations on M/M/1 queuing model
	Elementary Queuing Analysis: Queuing Networks and Case Studies
	Resource Management: Scheduling Fundamentals
	Resource Management: Basic Scheduling Algorithms
	Resource Management: Real-Time and Priority-Based Scheduling Algorithms
	Resource Management: State-Sensitive (I/O) Scheduling Algorithms
	Resource Management: Group Scheduling and Fairness
	Principles of Concurrency and Process Synchronization

	Mutual Exclusion: Synchronization Algorithms
	Mutual Exclusion: Semaphores and their implementation
	Mutual Exclusion: Producer/Consumer and Reader/Writer Problems
	Classical Synchronization: Dining Philosophers Problem
	Deadlock Management: From Detection to Avoidance
	Transactional Processing
	Distributed Systems: Network layers and Internet protocols
	Distributed Systems: Synchronization and Distributed Global Snapshot
	Distributed Systems: Miscellaneous Topics