

Wright State University

## CORE Scholar

---

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

---

Winter 2007

### CEG 434/634-01: Concurrent Software Design

Natsuhiko Futamura

*Wright State University - Main Campus*

Follow this and additional works at: [https://corescholar.libraries.wright.edu/cecs\\_syllabi](https://corescholar.libraries.wright.edu/cecs_syllabi)



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

---

#### Repository Citation

Futamura, N. (2007). CEG 434/634-01: Concurrent Software Design. .

[https://corescholar.libraries.wright.edu/cecs\\_syllabi/1205](https://corescholar.libraries.wright.edu/cecs_syllabi/1205)

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact [library-corescholar@wright.edu](mailto:library-corescholar@wright.edu).

CEG434/634  
Concurrent Software Design  
Winter 2007  
Syllabus

Time: Monday, Wednesday, 4:10 pm to 5:25 pm

Class Room: 204 Fawcett

Instructor: Dr. Natsuhiko Futamura

Office: 335 Russ Engineering Center

Email: [natsuhiko.futamura@wright.edu](mailto:natsuhiko.futamura@wright.edu)

<http://www.cs.wright.edu/~nfutamur/>

Phone: 775-5107

Office Hours: 3:00-4:00PM on Monday and Wednesday, and 3:30-5:00PM at my office at 335 Russ Engineering Center. Or, by appointment.

TA: Paul Bender ([bender.13@wright.edu](mailto:bender.13@wright.edu))

TA office hours: Mondays and Wednesdays 3:00-4:00pm. at RC326

Prerequisite: CS400, CEG433/633, Operating Systems.

Expected background:

Discrete mathematics, Data structure, C or C++,  
Programming experience in UNIX.

This course provides an introduction to concurrent program design in the UNIX environment. Classical problems of synchronization, concurrency, and their solutions are examined through the course projects and through readings on operating system design text book.

**Text books:**

Required:

Operating Systems Concepts 7<sup>th</sup> Ed. Silbershatz and Gavlin, Addison Wesley, 2005

Unix systems Programming: Communication, Concurrency and Threads.  
Robbinson and Robbins, Prentice Hall 2003

References: Interprocess Communications in UNIX: The Nooks and Crannies, 2<sup>nd</sup> Ed.  
Johns S. Gray, Prentice Hal 1998

**Exam schedule:**

Midterm: Tuesday, Feb 6, In class exam

Final exam      Tuesday , March 13, 5:45-7:45PM

**Grading:** The grades will be based on a midterm exam, final exam, and homework assignments and project. They have the following weight towards the overall grade.

Projects: 20%  
Homework: 20%  
Mid-term: 30%  
Final: 30%

Letter grades are given based on the overall score.

A - 80% or above

B - 70% - 79%

C - 60% - 69%

D - 50% - 59%

F - below 50%

The letter grades are not intended to be curved; however, I reserve the right to curve the final grades based upon the final point distribution.

A missed exam counts as a 0. The grade A indicates excellence: To receive an A, you must demonstrate a thorough knowledge of the material throughout the course.

There will be no grades of incomplete given except when documented emergencies have made it unable for the student to finish the course.

**Topics:** The topics covered in the course include the following:

Process management  
Process scheduling  
CPU scheduling  
UNIX I/O Inter-process communication  
Asynchronous events  
Client-Server computing  
Inter-process communication and sockets  
Process Synchronization ( critical sections, semaphores, etc)  
Threads,  
Deadlocks