

COMPUTER SCIENCE 501  
Computing for Science and Engineering  
FALL 2009  
COURSE INFORMATION  
August 28

**CS 590-003:** See page 4 for modifications pertaining to CS 590-003.

**Course description:** Computational concepts, tools, and skills for computational science and engineering: scripting for numerical computing, scripting for file processing, high performance computing, and software development. Project may be required. Credit in this course may not be used toward a graduate degree in Computer Science. Prerequisites: C S 159, MA 262.

**Role of course:** This is a core course in the CS&E Program. The purpose of the course is to expose students to computational concepts, tools, and skills useful for research in computational science and engineering, beyond what is learned in a first programming course (and basic mathematics courses). A related aim is to prepare students for other courses in CSE core by teaching CS material mostly unavailable in existing graduate level courses.

**Prerequisites:**

- mathematical knowledge and maturity of an MS student in the physical sciences. (It is not assumed that the student is in the physical sciences, only that he(/she) has this level of mathematics.) In particular, some familiarity with matrix algebra and elementary probability is expected.
- programming experience in C, C++, Java, or Fortran or extensive scripting experience; also, commensurate computer skills.

**Staff:**

<b>Instructor</b>	<b>Teaching Assistant</b>
Robert Skeel	Lei Zhao
Lawson 1205	Lawson B116G
(765)494-9025	496-9433
fax (765)494-0739	
skeel at cs dot	zhao54 at
Office hours:	
Mondays	10:00–11:00*
Tuesdays 1:30–2:00	
Wednesdays 2:30–3:00	
Thursday	10:00–11:00*, 1:30–2:30
Fridays	9:30–10:30

\* only those weeks that an assignment is due

**Web page:** The class web page is <http://www.cs.purdue.edu/homes/skeel/cs501.html>. This will be neglected in favor of Blackboard Vista, which will also be used for grade records and turning in homework.

**Textbook:** There is no suitably comprehensive textbook. Notes will be distributed having links to web resources. A good reference is Software Carpentry. Some useful books:

- Python in a Nutshell  
by Alex Martelli, O'Reilly, March 2003, ISBN: 0-596-00188-6,
- Learning Python, 2nd Edition  
by Mark Lutz, and David Ascher, O'Reilly, December 2003, ISBN: 0-596-00281-5,
- Python Scripting for Computational Science, 3rd Edition, 2nd Printing  
by Hans Petter Langtangen, Springer, 2008,
- The C Programming Language, 2nd Edition  
by Brian W. Kernighan and Dennis M. Ritchie, Prentice-Hall, 1988, ISBN 0-13-110362-8,
- An Introduction to Parallel Computing: Design and Analysis of Algorithms, 2nd Edition  
by Ananth Grama, George Karypis, Vipin Kumar, and Anshul Gupta Addison-Wesley, 2003, ISBN: 0-201-64865-2.

These books are on reserve in the Mathematical Sciences Library.

### Examination schedule

Wednesday, **Oct 7** 1st and 2nd exam, 1:30-2:20 pm,

Friday, **Nov 13** 3rd and 4th exam, 1:30-2:20 pm,

**Dec 14–19** 5th, 6th, and final exam, 2 hours

The lowest score of exams 1–6 will be dropped.

### Anticipated course outline for Weeks 1–5

*Theme I: scripting to get things done*

**Aug 24, 26, 28, 31** introduction to CS&E, Unix, software installation

**Sep 2, 4, 9, 11** files and text processing in Python

**Sep 14, 16, 18, 21, 25** mathematical computing in Python

*Theme IV: numerical computing*

**Sep 23** data error, error propagation

## Policies

- a. *Copying* is the *evident* use of outside material in your solution (anything not present in lectures and handouts). In the case of collaboration or use of outside written material, the score for an assignment may be discounted. For example, 3 students turning in identical correct answers each may receive as little as one third credit. The course staff reserve the right to determine the discount, if any. Use of your work by another will be assumed to be done with your consent unless there is evidence to the contrary.
- b. *Cheating* is the evident use of outside material in your solution *without including a written acknowledgment of the source*. (Citing sources is simply good scholarship.) This does not preclude some discussion when starting an assignment. However, the onus is on you to either acknowledge the collaboration or ensure that it is sufficiently limited so as not to be evident in your work. In practice this is enforced when the course staff believe that the collaboration goes beyond the sharing of ideas. but again the course staff reserve the right to make this judgment. *Cheating is penalized in accordance with university rules and the instructor's policy is to impose substantial penalties.*
- c. Excuses. Failure to attend an exam or failure to submit an assignment on time is recorded as a zero except when it is unavoidable because of some legitimate emergency (a medical excuse from a physician, or a death in the immediate family). In such a case it is desirable that the instructor be contacted before the exam or before the due date of the assignment.  
  
The teaching assistant may grant you an extension for an assignment, but he has the prerogative to refuse you.
- d. Requests for assignment regrades should be submitted to the TA within one week of it being returned to the class. If you remain dissatisfied after the regrade, you may appeal to the instructor.
- e. No extra work will be provided for students wishing to improve their grade. Also, final grades will not be changed after they are posted (except if they were incorrectly computed).

**Assignments** There are about six assignment sets, much of it requiring the writing of programs. Tentative due dates are Sep 11, Sep 25, Oct 16, Oct 30, Nov 20, and Dec 4. The plan is to drop the lowest score. There is no project this semester. We will use Python and C. There will be considerable computer use. Ideally, the student already has access to a multicore computer for his/her research and will use this for the assignments. The student will be expected to install the necessary (free) software on his/her desktop. As an alternative, accounts on an ITaP/RCAC multiprocessor are provided. One way or another, we will attempt to accommodate a variety of platforms. In any case, we will ask you what platform (hardware and system software) you are using.

## Grading

assignments	40%	A	80–100%	D	40–50%
in-class exams	27–36%	B	65–80%	F	0–40%
final-week exams	24–33%	C	50–65%		

Plus and minus grades will also be given. The grade boundaries may be lowered by as much as 10% in order to achieve a reasonable grade distribution. Also, there may be minor upward or downward adjustments of the grade boundaries.

**Plans for potential campus interruptions** In the event of a major campus emergency, course requirements, deadlines and grading percentages are subject to changes that may be necessitated by a revised semester calendar or other circumstances. Here are ways to get information about changes in this course: The primary method of communicating with students is Blackboard Vista. The course website might be used a backup, as might email.

COMPUTER SCIENCE 590C  
Computing for Life Sciences  
FALL 2009  
COURSE INFORMATION  
August 24

**Course description:** Computational concepts, tools, and skills for computational life sciences: scripting for numerical computing and file processing. Credit in this course may not be used toward a graduate degree in Computer Science.

**Prerequisites:** Math/CS/Stats required of undergraduates in biology at Purdue.

**To be omitted:** Any books or software dealing with C or parallel programming, e.g., Python for MPI.

**Examination schedule**

Wednesday, **Oct 7** final exam, 1:30-2:20 pm,

**Anticipated course outline** Topics for Weeks 1–5.

**Assignments** There are two assignment sets, much of it requiring the writing of programs. Tentative due dates are Sep 11 and Sep 25. We will use Python. There will be considerable computer use.

**Grading**

assignments	52%
in-class exam	48%

**ASSIGNMENT # 0** The following actions are required:

1. Send an email to the TA stating precisely what operating system you are using. In the case of Windows, state whether or not you will be using cygwin. If you will use more than one platform, so state.
2. Install items 1–6 below. Consult Section 1.3 of the class notes for assistance and advice.

## **Installation of Software**

The following is required:

1. a C compiler that supports OpenMP, e.g., version 4.2 or later from [www.gnu.org](http://www.gnu.org). Test with the program `ohello.c` in the class notes.
2. MPI. Presumably this is already installed. Test with the program `hello.c` in the class notes.
3. Python 2.5.4 or 2.6.2.
4. NumPy.
5. MPI for Python (`mpi4py`).
6. a plotting program, e.g., Matplotlib.

The following is recommended:

7. IPython.