

COMPUTER SCIENCE 50100
Computing for Science and Engineering
FALL 2016
COURSE SYLLABUS
Aug 22

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: {CS 15900 or 17700 or 15800} and MA 26200.

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. A related aim is to prepare students for other courses in CSE core by teaching CS material mostly unavailable in existing graduate level courses. The following CSE core/relevant courses are probably accessible after taking this course: CS 51501, CS 52500, CS 53000, CS 53500. The following are probably accessible without taking this course: CS 51400, CS 51500, CS 52000.

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 is expected.
- {programming experience in C, C++, Java, or Fortran} or extensive scripting experience; also, commensurate computer skills.

Instructor:

Robert Skeel
Lawson 1205
(765)494-9025

email rskeel

Office hours:

Mondays 1:30–2:00 pm
Tuesdays 2:00–2:30 pm
Fridays

Teaching Assistant:

Youhan Fang
TBD
TBD

email yfang

* 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

- **piazza** for announcements, handouts, and discussion; and
- **Blackboard Learn** for grade records and turning in homework.

Textbook: There is an optional textbook.

- **Big Practical Guide to Computer Simulations, 2nd Edition**
by Alexander K Hartmann, World Scientific, March 2015.

Also notes will be distributed having links to web resources. Other useful books:

- **Python in a Nutshell, 2nd Edition**
by Alex Martelli, O'Reilly, July 2006,
- **The C Programming Language, 2nd Edition**
by Brian W. Kernighan and Dennis M. Ritchie, Prentice-Hall, 1988, ISBN 0-13-110362-8,
- **Introduction to High Performance Scientific Computing**
by Victor Eijkhout, 2010.

A useful website is **Software Carpentry**.

Schedule exams

Thursday, **Oct 6**, 8:00-9:30 pm, PHYS 112, 1st and 2nd exam,

Thursday, **Nov 10**, 8:00-9:30 pm, PHYS 112, 3rd and 4th exam,

Dec 12–17 5th, 6th, and final exam, 2 hours.

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

Anticipated course outline

Part I: scripting to get things done

August Unix, software installation

September file/text processing and mathematical computing in Python

Part II: programming for performance

October C programming

November parallel computing: MPI, OpenMP

Part III: survey of other topics

December developing software,

Policies

- a. *Copying* is the *evident* use of outside material in your solution (anything not present in lectures and handouts). Outside material includes everything except material composed by the instructor. 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.¹ *Cheating is penalized in accordance with university rules and the instructor's policy is to impose substantial penalties.* **“Students . . . can report issues of academic integrity that they observe, either through the Office of the Dean of Students (purdue.edu/odos), call 765-494-8778 or email integrity@purdue.edu.”**
- 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 14, Sep 28, Oct 19, Nov 2, Nov 16, and Nov 30. The plan is to drop the lowest score. There is no project this semester. We use bash, Python 3, and C (C99 or C11). There will be considerable computer use. Ideally, the student already has access to a multicore computer for his/her research (having at least 2 cores and 2 hyperthreads/core for a total potential speedup of 4) 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 the ITaP/RCAC Scholar cluster are provided. One way or another, we will attempt to accommodate a variety of platforms.

Grading

¹In 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.

best 5 of 6 assignments	35%	A	85–100%	D	45–55%
best 5 of 6 exams	50%	B	70–85%	F	0–45%
final exam	15%	C	55–70%		

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 Piazza. The course website might be used a backup, as might email.

ASSIGNMENT # 0 (Required)

Due Wednesday, September 7 at 12:30 pm

1. (Required) Be reminded that the evident use of outside material in your solution without including a written acknowledgment of the source is *cheating*. Read all the course policies in the syllabus, and **state that you have done so**.

Send your solutions electronically You should submit one text file containing your response to item 1.

1. Log on to Blackboard at <http://www.itap.purdue.edu/tlt/blackboard/> (To use Blackboard, you have to have Java enabled for your web browser.)
2. Click on Course Content from the left panel. You will find the homeworks.
3. Go into a homework assignment, you will see a submission window on the page. Click the "Browse My Computer" Button. You will be able to submit the `.txt` file containing your homework answers there.

Installation of Software

The following is required:

1. a C compiler that supports OpenMP and variable dimension arrays, e.g., version 4.2 or later from www.gnu.org. Test with the program `ohello.c` in the class notes.
2. MPI. Test with the program `hello.c` in the class notes.
3. Python 3.5. Note that Python 3 is not backward compatible with Python 2.
4. NumPy 1.3.11.
5. MPI for Python (`mpi4py`) 2.0.
6. Matplotlib 1.5.

The following is recommended:

7. IPython 4.2.

Consult Section 1.3.1 and 1.3.2 of the class notes for assistance and advice.

If you are running Linux or OS X, you can use `ssh` and `scp` on `scholar`, or any other Unix machine, without entering passwords by doing the following: First, see whether you already have the file `~/.ssh/id_rsa.pub` on your computer. If not, create such a file by entering the following command on your machine

```
$ ssh-keygen -t rsa
```

and hitting return 3 times. Then, append the contents of this file to the file

`scholar:~/.ssh/authorized_keys`. Further information is available from <http://rcsg-gsir.imsb-dsgi.nrc-cnrc.gc.ca/documents/internet/node31.html>.