COS 397: Computer Science Capstone 1

Sudarshan S. Chawathe

University of Maine

Fall 2016

This course is the first of a two-course sequence designed to guide students in proposing the Capstone project in either an independent study, group project, or field experience format. The focus is on the early stages of project work, including finding a suitable topic and project advisor, investigating related work, and writing a thorough project proposal. The relevant skills are covered and practiced by studying a collection of classic and recent papers. The papers and topics studied vary by semester, partly based on discussions in the first few class meetings. An important aspect of this course is the integration of knowledge and skills acquired in separate courses in the rest of the curriculum.

News and Reminders:

- Please read the class newsgroup for timely announcements: `umaine.cos397` on NNTP server `creak.um.maine.edu`. Web interface to get started: `http://cs.umaine.edu/~chaw/news/`.
- The most recent version of this document may be found at `http://cs.umaine.edu/~chaw/cap1/`.
- Please use the PDF version of this document for printing and reference: `cos397.pdf`

Goals and Learning Objectives

Goals

- Integrate knowledge and skills acquired in other courses.
- Learn new material with attention to the learning process.
- Develop the ability to independently explore a topic by discovering, reading, and critiquing prior work.
- Improve our communication skills, with particular emphasis on written communication and, further, well-written programs.
- Practice the appropriate and ethical use of existing material of different kinds, such as source code, services, and documentation.
- Learn how to manage a self-directed project.

Learning Objectives

Students should be able to

- Develop effective learning strategies for continuing acquisition of knowledge and skills.
- Make effective use of the research literature.
- Understand and follow formal and informal standards of the field.
- Present their work in a public forum to their peers and others.

Prerequisites

The prerequisite for Capstone 1 is junior standing in Computer Science or permission of the instructor. Students should discuss this prerequisite with their academic advisors before seeking help elsewhere.

Contact Information

Class meetings:

Time: Tuesdays & Thursdays, 2:00 p.m.–3:15 p.m.
Location: Barrows Hall, Room 125.

Instructor: Sudarshan S. Chawathe

Office: Neville Hall, Room 224.
Office hours: (Please check for changes.)
Tuesdays and Thursdays: 3:15–4:30 p.m.
Phone: +1-207-581-3930.
Please avoid calling except for truly urgent matters.
Email: `sudarshan.chawathe@maine.edu`
Use email only for messages unsuitable for the newsgroup. (See below.) Please use only this email address and put the string *COS397* near the beginning of the Subject header of the message. *All other messages may be ignored.*
Web: `http://cs.umaine.edu/~chaw/`.

Teaching Assistant: Ezekiel Rhodes

Office: TBA
Office hours: TBA
Email: `Ezekiel_Rhodes@umit.maine.edu`

Online Resources

Class Web site: `http://cs.umaine.edu/~chaw/cos397/`
We will use the class Web site for posting assignments, readings, notes, and other material. Please monitor it.

Class Newsgroup: We will use the local USENET newsgroup `umaine.cos397` on the NNTP server `creak.um.maine.edu` for electronic discussions. If you are unfamiliar with USENET, you may find the Web interface at `http://cs.umaine.edu/~chaw/news/` useful as a quick way to get started. You may find further information on USENET at `http://en.wikipedia.org/wiki/Usenet`. The newsgroup is the primary forum for electronic announcements and discussions, so please monitor it regularly, and post messages there as well. Unless there is a reason for not sharing your question or comment, please *use the newsgroup, not email*, for questions and comments related to this course.

Class mailing list: *Please make sure you are on the class mailing list.* A sign-up sheet is circulated at the first class meeting. If you miss it, please contact me to get on the list. We will use this mailing list only for urgent messages because all other messages will go on the class newsgroup. I anticipate fewer than a dozen messages on this list over the semester.

Grading Scheme

Grade components: *Students are expected to complete and submit all assigned coursework in good faith; those who fail to do so may earn a failing grade, regardless of overall numerical score.*

component	% grade	
class participation	10	
classroom exercises	20	
project proposals (versions 1, 2, & 3)	50	(10 + 20 + 20)
final posters	20	

Class participation: Students are expected to contribute to learning by asking questions and making relevant comments in class and on the class newsgroup. Quality is more important than quantity. Disruptive activity contributes negatively (see policies).

Classroom exercises: Our work in the classroom will include a number of short group exercises, meant to solidify understanding of the concepts being discussed. One or more such exercises are likely to be part of most class meetings. The exercises will be graded primarily for effort, group work, and other contributions, and less so for simple correctness. Since attendance is not mandatory (see policies below), some low-scoring exercises will be dropped for each student. Please see me if you have concerns about the interaction of this component and the attendance policy. The exercises require students to complete reading and other assignments assigned as homework in earlier classes.

Homeworks: Homeworks include programming and non-programming ones, often mixed. No collaboration is permitted. You are encouraged to discuss the problems and solution strategies *at a high level*, but the final solution and details must be your individual work. If you are unclear on the boundary between permissible and non-permissible interactions in this regard, please ask me.

Project Proposals: The sequence of three project proposals serves to develop a systematic plan for a capstone project. The details are outlined in the guide for Capstone project proposals (Reading 1). Further details will follow in class.

Policies

Due dates: All due dates (and times) are strict (to the minute), as announced in class. If you believe your work was delayed by truly exceptional circumstances, let me know as soon as those circumstances are known to you and I will try to make a fair allowance. However, *the default is that you get a zero if you don't turn in the work on time.*

Attendance: Although I expect students to attend all class meetings, I will not be taking attendance. *If you miss a class meeting, you are responsible for catching up on the lost material, including any important announcements made in class, on your own.* If you have a valid reason for missing a class, let me know early and I will try to help you make up the class. There will be no make-up exams or exercises. Missed items earn zero credit. If you have a valid reason for missing a test, let me know as early as that reason is known to you and I will make a fair allowance (but there will be no make up exam in any case).

Classroom activities: This course is based on an *active learning format*, so effective classroom activities are critical to its success. Students are expected to contribute to their own learning and that of their classmates, and to devote 100% of their attention to these activities while in class. On a similar note, all electronic and other distractions (computers, phones, assorted gizmos, etc.) must be *completely silenced and put away* for the entire duration of the class. (Students who need any such devices for disability accommodations should follow the guidelines outlined below. Others who need any accommodation in this regard due to special circumstances should make advance arrangements with the instructor.) Students who violate these rules or otherwise cause distractions in class will be asked to leave with no warning; habitual violators will face disciplinary action.

Office hours: All students are encouraged to make use of both the instructor's and TA's office hours to further their learning, obtain assistance on homework assignments, obtain feedback on their class performance, etc. However, office hours are not to be used as a substitute for attending and participating in class meetings (see above). Similarly, assistance with homework assignments will be limited to what is appropriate based on fairness to all; students are expected to demonstrate substantial effort on the assignment before seeking assistance.

Make-up classes: I may have to reschedule a few classes due to my other professional commitments. I will make every attempt to minimize the number of such occurrences and to reschedule for a time that works for most students. Further, I will make sure no student is penalized by such occurrences.

Academic honesty (standard university wording): Academic dishonesty includes cheating, plagiarism and all forms of misrepresentation in academic work, and is unacceptable at The University of Maine. As stated in the University of Maine's online undergraduate Student Handbook, plagiarism (the submission of another's work without appropriate attribution) and cheating are violations of The University of Maine Student Conduct Code. An instructor who has probable cause or reason to believe a student has cheated may act upon such evidence, and should report the case to the supervising faculty member or the Department Chair for appropriate action.

Disabilities (standard university wording): If you have a disability for which you may be requesting an accommodation, please contact Ann Smith, Director of Disabilities Services, 121 East Annex, 581-2319, as early as possible in the term.

Special circumstances (standard university wording): In the event of an extended disruption of normal classroom activities, the format for this course may be modified to enable its completion within its programmed time frame. In that event, you will be provided an addendum to the syllabus that will supersede this version.

Readings

This list will be revised and annotated as the semester progresses to reflect, in particular, the topics and papers selected based on class discussions. (The Assigned section below will grow.)

Assigned

1. Sudarshan S. Chawathe. Capstone project proposals—suggestions for deeper explorations. Department of Computer Science, University of Maine. `http://cs.umaine.edu/`, January 2010.

Others

1. Bruce Barnett. Sed—an introduction and tutorial. `http://www.grymoire.com/Unix/Sed.html`, July 2008.
2. Online resources describing *GNU make*.
3. Ken Thompson. Reflections on trusting trust. *Communications of the ACM*, 27(8):761–763, August 1984.
4. Online tutorials on *git* and version control.
5. Derrick Coetzee. An efficient implementation of Blum, Floyd, Pratt, Rivest, and Tarjan's worst-case linear selection algorithm. `http://moonflare.com/`, January 2004.
6. Rob Weir. Doing the Microsoft shuffle: Algorithm fail in browser ballot. `http://www.robweir.com/`, February 2010.
7. Jon Bentley. Little languages. *Communications of the ACM*, 29(8):711–721, August 1986.
8. Michael E. Lesk and Eric Schmidt. Lex—a lexical analyzer generator. In Andrew G. Hulme and M. Douglas McIlroy, editors, *UNIX Vol. II: research system*, pages 375–387. W. B. Saunders Company, Philadelphia, Pennsylvania, 10th edition, 1990.
9. Stephen C. Johnson and Ravi Sethi. In Andrew G. Hulme and M. Douglas McIlroy, editors, *UNIX Vol. II: research system*, chapter Yacc: a parser generator, pages 347–374. W. B. Saunders Company, Philadelphia, Pennsylvania, 10th edition, 1990.
10. Jon L. Bentley and M. Douglas McIlroy. Engineering a sort function. *Software–Practice and Experience*, 23(11):1249–1265, November 1993.
11. Timothy Furtak, José Nelson Amaral, and Robert Niewiadomski. Using SIMD registers and instructions to enable instruction-level parallelism in sorting algorithms. In *Proceedings of the 19th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 348–357, San Diego, California, 2007.
12. Bingsheng He, Ke Yang, Rui Fang, Mian Lu, Naga K. Govindaraju, Qiong Luo, and Pedro V. Sander. Relational joins on graphics processors. In *Proceedings of the 28th ACM International Conference on Management of Data (SIGMOD)*, Vancouver, Canada, June 2008.

13. Naga K. Govindaraju, Jim Gray, Ritesh Kumar, and Dinesh Manocha. GPUTeraSort: High performance graphics coprocessor sorting for large database management. In *Proceedings of the 26th ACM International Conference on Management of Data (SIGMOD)*, Chicago, Illinois, July 2006.
14. Daniel Cederman and Philippas Tsigas. A practical quicksort algorithm for graphics processors. Technical Report 2008-01, Department of Computer Science and Engineering, Chalmers University of Technology and Göteborg University, Göteborg, Sweden, 2008.
15. Sang-Won Lee and Bongki Moon. Design of flash-based DBMS: an in-page logging approach. In *Proceedings of the 27th ACM International Conference on Management of Data (SIGMOD)*, pages 55–66, Beijing, China, June 2007.
16. Gilad Bracha. Generics in the Java programming language. Tutorial. `http://java.sun.com/`, July 2004.
17. Mark C. Hamburg. Two tagless variations on the Deutsch-Schorr-Waite algorithm. *Information Processing Letters*, 22:179–183, 1986.
18. Martin E. Hellman. An overview of public-key cryptography. *IEEE Communications Magazine*, 50(5):42–49, May 2002. Originally published in 16(6), November 1978.
19. Jon Bentley and Don Knuth. Programming pearls: Literate programming. *Communications of the ACM*, 29(5):364–369, May 1986.
20. Jon Bentley, Don Knuth, and Doug McIlroy. A literate program. *Communications of the ACM*, 29(6):471–483, June 1986.
21. Paul E. Black. Dictionary of algorithms and data structures. `http://www.nist.gov/dads/`, September 1998.
22. Nadathur Satish, Mark Harris, and Michael Garland. Designing efficient sorting algorithms for manycore GPUs. In *Proceedings of the 23rd IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 1–10, Rome, Italy, May 2009.
23. Lloyd Allison. Suffix trees. `http://www.allisons.org/ll/AlgDS/Tree/Suffix/`, 2008.

Exercises and Notes

Material will appear here as we move along the semester. It may be useful to refer to the homeworks and tests from the previous session: `http://cs.umaine.edu/~chaw/201409/cos397/`.

- Class Exercises:
  - CE 01: `hwq/ce01.pdf`.
  - CE 02: `hwq/ce02.pdf`.
  - CE 03: `hwq/ce03.pdf`.
  - CE 04: `hwq/ce04.pdf`.
  - CE 05: `hwq/ce05.pdf`.
  - CE 06: `hwq/ce06.pdf`.
  - CE 07: `hwq/ce07.pdf`.
  - CE 08: `hwq/ce08.pdf`.
  - CE 09: `hwq/ce09.pdf`.
  - CE 10: `hwq/ce10.pdf`.
  - CE 11: `hwq/ce11.pdf`.
  - CE 12: `hwq/ce12.pdf`.
  - CE 13: `hwq/ce13.pdf`.

Submission Instructions

All electronic submissions must be made using the upload interface at `http://cs.umaine.edu/~chaw/u/`. *Electronic submissions in all other forms, such as email or physical media, will be discarded and receive no credit.*

Uploaded *files must be named* following this template:

`cos397-pr01-`*`Lastname`*`-`*`Firstname`*`-`*`N`*`.jar`

The substrings `pr01` and `jar` are replaced by others depending on the material being submitted and N is an arbitrary 4-digit number, such as 4231. Multiple submissions, within reason, may be made by selecting different values of N.

If your upload is successful, you will be presented with a confirmation Web page similar to the following sample. You should record the reported MD5 checksum and timestamp.

```
SUCCESS: Please note the following for your records.

Successfully saved cos397-pr01-Aardvark-Alice-1389.jar.
MD5 checksum: 09ee098b83d94c7c046d6b55ebe84ae1
Timestamp: 2016-09-13 13:32:34
```

If you do not see something very similar then your submission is unsuccessful.

If (and only if) there are unexpected problems and you are unable to submit your work as above, then you should save your file on your own computer (with some backups), compute its MD5 checksum using the md5sum utility on Unix-like systems (or other similar tools), and submit the file name, time stamp, and MD5 checksum (only, not the file itself) by email with a suitable Subject header.