COS 498: Capstone Projects in Computer Science

Sudarshan S. Chawathe

University of Maine

Spring 2008

The two Capstone courses provide milestones and guidance for students' work on their Capstone projects. Capstone 1 is a 1-credit course to be taken in the Spring semester of a student's junior year. Capstone 2 is a 2-credit course to be taken in the Spring semester of a student's senior year. These courses share class meetings. The bulk of the work for both courses (i.e., the project work itself) is to be done outside the classroom. For more information on the courses, please refer to the *Capstone Guide* in the Readings section below.

News and Reminders:

- Look for material in the *Assignments, Tests, and Notes* section.
- Midterm 2 solutions posted.
- Answers to several questions on *Engineering a Sort Function* posted.
- Source code for the GNU glibc implementation of qsort posted.
- Notes on *developing version 2 of the proposal* posted.
- Notes on *Engineering a Sort Function* posted.
- Please read the newsgroup for timely announcements.
- Class newsgroup: Local group `umaine.cs.capstone` on NNTP server `news.cs.umaine.edu`. Web interface to get started: `http://cs.umaine.edu/~chaw/news/`.
- Please use the PDF version of this document for printing and reference: `capstone.pdf`

Goals

- Practice the appropriate and ethical use of existing material of different kinds, such as source code, services, and documentation.
- Gain experience in contributing to the body of knowledge.
- Gain experience in conducting and documenting experimental studies of programs.
- Improve our programming skills, with attention to software engineering principles.
- Improve our communication skills, with particular emphasis on written communication and, further, well-written programs.
- Learn how to manage a self-directed project.

Prerequisites

Students should discuss the following prerequisites with their *academic advisors* before seeking help elsewhere. Students with any special requests in this regard must address them to the department chair, with the support of their academic advisor.

Capstone 1: The prerequisites for Capstone-1 are junior standing and permission of the department chair. Students who have a strong background and wish to take Capstone-1 early (before they have official junior standing) should confer with their academic advisors.

Capstone 2: The three prerequisites for Capstone-2 are Capstone-1, senior standing, and permission of the department chair. *Permission to register will be granted only to those students who have made enough progress in their project work to indicate a high likelihood of timely project completion.* This assessment will be made by the department chair, in consultation with the faculty. A key factor is the recommendation from the project advisor with additional input from the academic advisor.

Contact Information

Class meetings:

Time: Tuesdays & Thursdays, 8:20–9:20 a.m.
Note: We will *typically* meet on Tuesdays only. However, *students should make sure their schedule accommodates both the Tuesday and the Thursday time-slot*, as we will occasionally schedule additional meetings on Thursdays.
Location: Neville Hall, Room 406.

Instructor: Sudarshan S. Chawathe

Office: Neville Hall, Room 224.
Office hours: (Please check for changes.)
Tuesdays 8:00–8:20 a.m., 9:20–10:00 a.m., 2:00–3:00 p.m., 5:30–6:00 p.m.
Phone: +1-207-581-3930.
Email: `chaw@cs.umaine.edu`
Use email only for messages unsuitable for the newsgroup. (See below.) Please put the string *Capstone* near the beginning of the Subject header of your messages to me.
Web: `http://cs.umaine.edu/~chaw/`.

Online Resources

Class Web site:
`http://cs.umaine.edu/~chaw/capstone/`
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.cs.capstone` on the NNTP server `news.cs.umaine.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.

Schedule

I typically summarize scheduling matters at the beginning and end of each class. An approximate schedule appears in Figure 1. Please use it only as a rough guide to plan your studies. Do *not* use it to schedule travel or other events. If you need a definite answer on when something will or will not occur, you should check with me.

Grading Scheme

Grades: Grades are based on class participation (10%), two mid-term exams (10% each), two preliminary project proposals (10% and 20%), and a final project proposal (40%).

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. Please make sure all disruptive devices are disabled while in class. If you have a good reason for wanting to be disturbed in class, please contact me to make the appropriate arrangements.

Exams: All exams are *open book, open notes*. The exams in this course will consist of brief questions on topics discussed in class or assigned as reading. You are free to bring with you any resources that you find useful. However, no communications are permitted other than between students and me. The use of computers during exams is strongly discouraged, but brief use is permitted *provided it does not cause a disturbance*. You may use the Internet, but only as a library to look up material you may find useful. As above, check with me if you are unclear on what is permitted. The exams are designed to require no equipment other than a pen and paper.

Project Proposals: Developing a solid project proposal that will serve as the foundation for

#	Date	Material
1	*01-15*	Introduction [1].
2	*-22*	Random structures; J2ME projects [2].
3	*-29*	Student presentations.
4	*02-05*	Compilers and trust [3].
5	*-12*	First midterm exam; Project proposal version 1 due.
6	*-19*	Garbage collection; tagless Deutsch-Schorr-Waite [4].
7	*-26*	
		Spring Break March 1–16.
8	*03-18*	Student presentations.
9	*-25*	Quicksort implementation [5].
10	*04-01*	Public-key cryptography [6].
11	*-08*	Review & student presentations.
12	*-15*	Second midterm exam; Project proposal version 2 due.
13	*-22*	Review & student presentations.
14	*-29*	Review & student presentations.
15	**05-05**	**8:00 a.m.: Final project proposals due ("final exam").**

Figure 1: **Approximate** *Schedule. Additional meetings, scheduled for Thursdays by default, are not shown. The numbers in square brackets refer to the readings.*

continuing work on a Capstone project is a primary objective of this course. This emphasis is reflected in the weights assigned to the two preliminary and final project proposals. Further details will follow.

Policies

Due dates: All due dates are strict, 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.*

Special needs: If you have special needs of any kind, including, but not limited to, disabilities, absences due to participation in sports or other activities, etc., please contact me *as soon as the need is known to you* and I will try to accommodate them as much as possible.

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 making up the lost material. 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.

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: I expect you to hold yourselves to the highest standards of academic honesty. Please take this point very seriously. If you are not sure if something is permitted, check with me. *All help you receive, even if permitted, must be prominently noted in all work you submit.* Erring on the side of giving too much credit is far better than the alternative. Plagiarism and other forms of cheating will result in very stiff penalties (including, but not limited to, an F grade in the course and further disciplinary action from the university).

Readings

1. Sudarshan S. Chawathe. Capstone projects and courses: A guide for Computer Science majors. Department of Computer Science, University of Maine. `http://cs.umaine.edu/`, September 2007.

2. Gilad Bracha. Generics in the Java programming language. Tutorial. `http://java.sun.com/`,

July 2004. This tutorial is highly recommended for everyone, but especially for those planning to use Java for their projects. The concepts explained here are essential for making good use of generics in Java and it is very painful to learn them the hard way (e.g., while debugging your code).

3. Ken Thompson. Reflections on trusting trust. *Communications of the ACM*, 27(8):761–763, August 1984.

4. Mark C. Hamburg. Two tagless variations on the Deutsch-Schorr-Waite algorithm. *Information Processing Letters*, 22:179–183, 1986.

5. Jon L. Bentley and M. Douglas McIlroy. Engineering a sort function. *Software–Practice and Experience*, 23(11):1249–1265, November 1993.

6. 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.

7. Derrick Coetzee. An efficient implementation of Blum, Floyd, Pratt, Rivest, and Tarjan's worst-case linear selection algorithm. `http://moonflare.com/`, January 2004.

8. Jon Bentley and Don Knuth. Programming pearls: Literate programming. *Communications of the ACM*, 29(5):364–369, May 1986.

9. Jon Bentley, Don Knuth, and Doug McIlroy. A literate program. *Communications of the ACM*, 29(6):471–483, June 1986.

10. Paul E. Black. Dictionary of algorithms and data structures. `http://www.nist.gov/dads/`, September 1998.

Assignments, Tests, and Notes

1. First midterm exam: `hwq/mt01.pdf`.
   Sample solutions: `p/mt01s.pdf`.

2. Notes on *Engineering a Sort Function*: `n/enggsort.pdf`.
   Sample solutions: `p/enggsorts.pdf`.

3. GNU C library implementation of qsort: `n/qsort.html`.

4. Notes on developing version 2 of the proposal: `n/cpguide.pdf`.

5. Second midterm exam: `hwq/mt02.pdf`.
   Sample solutions: `p/mt02s.pdf`.