

2009 Student Faculty Conference
Committee on Computer Science
June 14, 2009

Contents

1	Introduction	3
1.1	Membership	3
2	Issues	3
2.1	Course Offerings	4
2.1.1	Background	4
2.1.2	Long-Term Recommendations	5
2.1.3	Short-Term Recommendations	5
2.2	Introductory Courses	6
2.2.1	Background	6
2.2.2	Long-Term Recommendations	7
2.2.3	Short-Term Recommendations	7
2.3	Teaching Quality	7
2.3.1	Background	7
2.3.2	Long-Term Recommendations	8
2.3.3	Short-Term Recommendations	8
2.4	Research Focus	8
2.4.1	Background	8
2.4.2	Long-Term Recommendations	8
2.4.3	Short-Term Recommendations	8
2.5	Industry Focus	8
2.5.1	Background	8
2.5.2	Long-Term Recommendations	9
2.5.3	Short-Term Recommendations	9
3	The SFC Panel	9
4	Current Status of Recommendations	9
A	Survey Results	9
A.1	Questions by the Undergraduate Research Committee	9
A.2	Questions by the CS Committee	11
B	Historic Course Offerings	16
C	New Course Proposals and Descriptions	18

1 Introduction

The 2009 Student Faculty Conference Committee on Computer Science was composed of undergraduates and faculty charged with evaluating the undergraduate computer science option at Caltech and with making recommendations to improve it. From December 2008 through April 2009, the committee's members held discussions to identify the option's strengths and weaknesses, to develop short-term solutions, and to make long-term recommendations for the option.

The computer science option was first offered to the class of 2004. While the 2007 Student Faculty Conference held a joint committee between electrical engineering and computer science, this committee was devoted entirely to the computer science option alone. As of February 3rd, 2009, there were 71 sophomores, juniors, and seniors declared in computer science.

The committee held its first meeting in December 2008. After discussion over the group's email list, a survey was formed and submitted to the ARC for distribution over Winter Break. After the survey results were made available to the committee in mid-February, the committee met once to twice per week in order to discuss its areas of concern.

1.1 Membership

- **Student Chair:** Chris Kennelly, 2011
- **Undergraduate Members:**
 - Cheng William Hong, 2009
 - Andy Matuschak, 2010
 - Matthew Maurer, 2010
 - Alex Roper, 2009
 - Karthik Sarma, 2011
- **Faculty Members:**
 - Al Barr, Professor of Computer Science
 - K. Mani Chandy, Simon Ramo Professor and Professor of Computer Science
 - Mathieu Desbrun, Associate Professor of Computer Science, Graduate Option Representative for Computer Science
 - Donnie Pinkston, Lecturer in Computer Science
 - Chris Umans, Associate Professor of Computer Science
 - Mike Vanier, Lecturer in Computer Science
 - Adam Wierman, Assistant Professor of Computer Science

2 Issues

The committee met in December 2008 in order to identify possible issues of discussion for the Student Faculty Conference. These issues were then distilled into a survey which went out to undergraduates in the computer science option. Out of 71 students declared in the option as of February 2009, 37 responses were collected from 10 sophomores, 12 juniors, and 15 seniors. The survey results and questions are included in Appendix A.

2.1 Course Offerings

2.1.1 Background

The undergraduates on the committee brought course offerings as their primary concern to the preliminary SFC meeting. Within the computer science option, students are required to complete a three-term project sequence, an approved, independent laboratory project spanning at least two terms, or a faculty-supervised thesis. Discussion amongst students and faculty suggests that project sequences are the most frequent mode by which undergraduates in the option fulfill this requirement. According to Mary Morley, Caltech's Registrar:

- In 2008, 23 students graduated in the computer science option:
 - 1 completed EE/CS 80 (Senior Thesis)
 - 10 completed CS134 (Computer Systems and Compilers)
 - 7 completed CS17x (Graphics)
 - 3 completed CS129 (Information and Complexity)
 - 2 completed CS141 (Distributed Systems)
- In 2009, 25 students are expected to graduate:
 - 2 completed CS81 (Undergraduate Laboratory in Computer Science)
 - 1 completed CS90 (Undergraduate Research in Computer Science)
 - 9 completed CS141 (Distributed Systems)
 - 5 completed CS136 (Programming Languages)
 - 4 completed CS17x (Graphics)
 - 4 completed CS129 (Information and Complexity)

In the 2008-09 academic year, only CS145 (Networking), CS17x (Graphics), and CS181 (VLSI Design) were offered in full. Additionally, undergraduates brought concerns to the committee that increased breadth, depth, and availability of electives was desired. Thus, breadth and availability are of utmost concern in the option.

The survey to undergraduates asked for which areas of computer science students were interested in. Additionally, CS undergraduates were asked to identify areas which were under or overrepresented in the option's course offerings. Nearly all areas of computer science research were considered underrepresented with the exception of computer graphics (See Table A.8). The top areas of interest were:

- AI/Learning Systems - Presently represented by CS156ab (Learning Systems).
- Algorithms - Presently represented by CS38 (Introduction to Algorithms), a CS core requirement, CS138abc (Computer Algorithms), CS150 (Probability and Algorithms). CS138 has been offered infrequently over the past several years. Survey results suggest that coverage of algorithms is desired yet considered underrepresented.
- Theory - Presently represented by CS21 (Decidability and Tractability), a CS core requirement, and CS151 (Complexity Theory). Similar to algorithms, computational theory is desired yet considered underrepresented.

- Graphics - Presently represented by CS/CNS 171 (Introduction to Computer Graphics Laboratory), CS/CNS 174 (Computer Graphics Projects), CS 175 (Geometric Modeling), CS 176 (Introduction to Computer Graphics Research), and CS 177 (Discrete Differential Geometry).
- Systems - Presently represented by CS24 (Introduction to Computer Systems), a CS core requirement, and CS134abc (Computer Systems).
- Languages - Historically, coverage of programming languages has been developed in CS136abc (Programming Languages Laboratory), a three-term project sequence. With the departure of Professor Jason Hickey from the department at the end of the 2007-08 academic year, this course's future is uncertain.

Full results are listed in Table A.7 and Table A.8.

A list of course offerings, including project sequences, is included in Appendix B.

2.1.2 Long-Term Recommendations

From the observations made, the committee recommends:

- That the department add new faculty members in order to increase the availability of existing courses listed in the catalog and to expand the breadth of courses available. During faculty searches, consider adding faculty in areas which are underrepresented. In the midst of the recession, budgetary issues currently constrain the department in its ability to appoint new faculty and lecturers.
- That the institute create an Information Science Technology division to house CS and related departments within the existing Engineering and Applied Sciences division. The creation of the division would allow computer science and its related disciplines to be adequately represented on an Institute-wide basis.
- That faculty continue to monitor of undergraduate interests and needs when determining course offerings.

2.1.3 Short-Term Recommendations

The committee recommended changes which could go into effect immediately. Relevant changes to the catalog were submitted to the Curriculum Committee for approval.

Additionally, the department should consider whether offering course materials online with recorded lectures or problem solving sessions in lieu of normal lectures would allow for more courses to be offered more often. These considerations must be made with extreme care. Student survey results were mixed (Table A.2) and members of the committee recognize that not all areas will be suitable for lecture-less formats.

- **AI/Learning Systems:** A new project sequence will be offered, CS/CNS/EE 159 (Projects in Machine Learning and AI). CS159 will be a one-term course which complements the existing CS156ab (Learning Systems) offering and two new one-term courses, CS154 and CS155. CS154 (Artificial Intelligence) will be first offered in the 2010-11 academic year. CS155 (Probabilistic Graphical Models) will be offered second term in the 2009-10 academic year. Successful completion of CS159 as well as either CS156ab or CS154/155 will satisfy the project sequence requirement of the CS option. An additional machine learning class, CS255 (Special Topics in Machine Learning) is planned for the 2009-2010 academic year.

- **Algorithms:** The first term of CS138 (Computer Algorithms) is being offered in the current (SP08-09) term by a postdoc in IQI (The Institute for Quantum Information). The committee does not have a proposal for offering the remainder of the sequence in the near future.
- **Languages:** The committee could not find a way of offering CS136 (Programming Languages Laboratory) in the near future.
- **Networking:** CS145abc will be refactored into three, one-term, sequential courses, CS143-145. CS143 will be entitled “Communication networks.” CS144 will be entitled “Ideas behind the web.” CS145 will provide for an a student project to complete the sequence in fulfillment of the undergraduate graduation requirements. Additionally, two new networking electives are expected for the next academic year, CS/EE 146 (Advanced networking) and CS/EE 147 (Network performance analysis). In the 2010-2011 academic year, a graduate-level course, CS/EE 245 (Special topics in networking) is planned.
- **Other:** EE/Ma 127 (Error-Correcting Codes) will be cross-listed with Computer Science. ME 131 (Advanced Robotics: Manipulation and Sensing) will be cross-listed as well.
- **Systems:** The committee proposed splitting CS134abc into two new, one-term courses, CS134 (Computer Systems) and CS135 (Compilers). Computer Systems is being offered in the 2008-09 academic year as CS134. In the short-term, a faculty member will need to be found in order to provide for a compilers offering. The committee believes that the split into two, independent courses will alleviate some of the burden associated with project courses, allowing the parts to be offered more frequently than the whole would be in the status quo.
- **Thesis and Research:** The committee proposed offering an introductory pizza course to the option, targeted primarily towards freshman and sophomores. The course would allow students to find out areas of current research amongst faculty.
- **Theory:** A new current topics in computer science theory course will be offered in the next academic year.

2.2 Introductory Courses

2.2.1 Background

The committee considered introductory courses in the computer science option such as CS1, CS2, CS3, and CS11. CS1 and CS2 are required for the CS option. These introductory courses are frequently required or recommended for other options. In the 2008-2009 catalog, the EAS option, Geophysics, and Mechanical Engineering allow undergraduates to satisfy requirements by taking some combination of these courses. Additionally, the upcoming undergraduate Bioengineering option can be partially satisfied by these courses.

Given the wide range of students taking these courses, it is key that they remain accessible to all students and not merely students in computer science. Additionally, concerns arise that with the wide diversity in programming experience for incoming freshmen, catering to these groups simultaneously is difficult and leads to worse outcomes. Inexperienced programmers are overwhelmed; experienced programmers are not challenged.

The CS1 curriculum—using Scheme as the language of choice—can also discourage new programmers from taking the course. For students outside of the option, required to take a sufficient number

of units of programming courses, they may often enter into CS2 and CS11, unprepared without independent programming experience or experience from CS1. On a related note, MIT removed Scheme from its introductory curriculum in computer science in the 2007-08 academic year.

Other concerns arose from the general curriculum of these courses. In conjunction with concerns regarding curriculum quality in upperlevel courses, the need to stress software engineering concepts was evident.

2.2.2 Long-Term Recommendations

The committee recommends monitoring the progress of the revised and expanded introductory programming curriculum.

2.2.3 Short-Term Recommendations

The committee, after discussion with the entire CS faculty, proposed modifying the CS curriculum to require three CS courses, CS1, 2, and 4.

- **CS1:** The course will be structured as an introductory programming course suitable for first-time programmers. This course will be a prerequisite for additional programming courses such as CS2, 4, and 11.
- **CS2:** The committee recommends modifying the CS2 curriculum as to provide for a survey of concepts and languages which were not considered in CS1 and CS4.
- **CS4:** This programming course will be required for CS majors and geared towards experienced programmers.

The full course descriptions for CS1 and CS4 are available in Appendix C. Continued discussion between students and faculty from the end of this student faculty conference up to the winter 2009-10 academic term will be directed towards revamping the CS2 curriculum.

Additionally, the committee recommended reconsidering the CS1 and CS2 placement exams. In the past, students have complained about the disparity between the placement exams and the classes for which the exams were testing for. Restructured placement exams will allow experienced programmers to elect to take different courses. Placement for existing students rather than merely incoming freshmen is recommended as well.

The committee believes that by imposing a prerequisite of CS1 or instructor permission, fewer students will use the CS11 tracks as an opportunity to learn their first programming language. By culling the number of first-time programmers in the class, the committee believes that it will reduce complaints about the uniting of CS11.

2.3 Teaching Quality

2.3.1 Background

The SFC survey brought up concerns about the teaching quality of some courses.

Other concerns were raised by students in the survey regarding the inclusion of software engineering concepts in project sequences. As noted in Section 2.1.1, many students will complete project courses to satisfy graduation requirements. Yet the introductory programming curriculum leaves students often ill-prepared for the process of working in a group to complete a software project. Much of the opportunity for incorporating these concepts into the project sequences are lost.

2.3.2 Long-Term Recommendations

The committee recommends the formation of an informal, BUSAC-like committee of undergraduates and faculty in order to continue the work of SFC. Ideally, the committee would be able to meet once per term in order to discuss issues as they arise.

2.3.3 Short-Term Recommendations

The committee recommended the inclusion of software engineering concepts in courses which involve programming at all levels. Key concepts include:

- Documentation
- Version Control
- Build Tools

2.4 Research Focus

2.4.1 Background

During preliminary discussions, faculty members expressed concern that the thesis option was underutilized. Additionally, empirical observations suggest that many freshmen and sophomores in the option frequently take SURFs in other departments or at JPL.

2.4.2 Long-Term Recommendations

Further consideration of the thesis option remains. Faculty should advertise the thesis more widely and be available to advise students interested in completing a thesis. Additionally, the label of “Senior Thesis” should be reconsidered as it possible for juniors to complete a thesis.

Additional work remains in furthering the availability of CS research and SURF opportunities to undergraduates, particularly freshmen and sophomores.

2.4.3 Short-Term Recommendations

A new course (CS9) similar to pizza courses in other majors should be offered during the first or second term in order to allow prospective and current CS majors the opportunity to find out about research in the department.

2.5 Industry Focus

2.5.1 Background

Surveys of graduating seniors demonstrates that historically a large fraction of undergraduates in the option will leave to go to industry. Survey results strongly suggested that students felt it was imperative to offer courses in algorithms, systems, and compilers in order to prepare students adequately. Absent these courses, some students indicated that they were disadvantaged by the undergraduate CS option in competing for full time positions and internships.

2.5.2 Long-Term Recommendations

As part of monitoring course offerings in order to ensure enough breadth in the option to avoid over-concentration in course offerings, future consideration should be made as to ensure the department is responsive to industry expectations.

2.5.3 Short-Term Recommendations

The revival of CS134 for the SP08-09 term offers exposure to systems beyond the scope of CS24. Nevertheless, fundamental concerns remain about the future of CS135 (The proposed compilers course, see Section 2.1) and CS138 (Computer Algorithms).

3 The SFC Panel

On April 6, 2009, the committee organized a breakout session following the main, general-interest section of the earlier part of the day. The session was well attended by students as well as by faculty throughout the department.

The discussion focused on three issues:

- **Course Offerings:** Audience members were satisfied with the progress towards more course offerings made by the committee. Nevertheless, funding remained as the largest impediment to offering courses in key areas currently lacking faculty coverage. While it is desirable to offer CS135 (The proposed compilers course) in the 2009-2010 academic year, funding to hire an instructor is necessary and currently lacking. Web-based courses were recognized as a possible solution for some areas, but audience members recognized that the courses, even with problem solving sessions, would have a different dynamic. Additionally, for some fields, audience members, as well as the committee, believe that web-based courses would not be satisfactory.
- **Double Majors:** Some discussion concerned double majoring. With vague Institute policy, applying clear and consistent policy is difficult inside the option when considering petitions for a double major. Students asked for further clarification and documentation in order to provide a clear delineation of what was expected of students intending to double major.
- **Course Feedback:** Audience members suggested having more ombudsmen for classes. Many CS courses lack ombudsmen, but some students indicated a desire to have them. Additionally, suggestions were made that CS faculty meet for lunch on a regular basis (perhaps once per month) in order to encourage student-faculty interaction.

4 Current Status of Recommendations

The proposed changes to the curriculum and course offerings were approved by the CS faculty in mid-March. The faculty submitted these proposals to the Curriculum Committee for inclusion in the 2009-2010 catalog.

A Survey Results

A.1 Questions by the Undergraduate Research Committee

Table A.1: Which of these describes your search for research experience at Caltech? (Multiple responses permitted)

Count	Response
21	Sought a project and found one I liked
5	Sought and found a project that I didn't really like
6	Sought a project but did not find one that I wanted to take
6	Did not seek research

Table A.2: Which of the following describes your work? (Multiple responses permitted)

Count	Description
6	Menial
20	Interesting
15	Useful
2	Lab Assistant
13	Independent
10	Incomplete
5	Completed
13	Skills training

Table A.3: Which of these describes your research experience at Caltech? (Multiple responses permitted)

Count	Description
22	Summer research
5	Research within curriculum (units required for graduation)
7	Research outside curriculum (For pay, work-study, units not required for graduation, or no compensation)

Table A.4: Are you satisfied with your research experience at Caltech?

Count	Response
14	Yes
12	Neutral
5	No

Table A.5: Which of these describes your interaction with professors at Caltech? (Multiple responses permitted)

Count	Description
7	Regular casual interaction
10	Interaction through events (seminars, option teas, etc.)
24	Interaction mainly through classes
10	Never meet/talk to very much

A.2 Questions by the CS Committee

Are you happy at Caltech as a CS major?

- 24 - Yes
- 13 - No

If not, why?

- Course offerings are lacking and are limited to a few areas.
- Online course offerings are not a solution to our problems.
- We need more faculty members in the department.
- CS curriculum lacks preparation for software engineering to ready students for large projects and industry.
- There is no structure to which courses are going to be offered and which ones students should take.

How satisfied are you with the CS curriculum (i.e. intro courses, advanced courses – please give us specific comments about individual courses)?

- There is “no good systems course.” Without a systems professor, the major lacks legitimacy.
- Courses which should be offered *every year* (Systems and Advanced Algorithms) are only offered every other year, at best.
- Depth is available in only a few fields (Graphics and machine learning) when other fields are underrepresented (Algorithms, systems, languages, theory, and compilers)
- “I ran out of advanced CS classes I was interested in.”
- Introductory CS courses (like CS1 and CS2) are not indicative of what CS is like.

Do you think the CS curriculum should be modified to have predefined 'tracks' for different areas in CS?

- 10 - Yes
- 21 - No
- 5 - Don't know

Are CS courses distributed well across terms? Are there any courses with significant time conflicts? Please give specific examples.

- Ma/CS 6b conflicts with CS21
- CS21, CS24, and CS38 should be in different terms.
- It shouldn't be necessary to timeconflict with Core to take CS24 and CS38 freshman year.
- Too few courses were offered this year second term.

Table A.6: Which of the follow do you think needs to be improved? (3 responses maximum)

Count	Areas of potential improvement
18	How students find projects
16	How students find mentors
8	Student-mentor interaction
11	Unit space in curriculum for research
8	Thesis opportunities

Table A.7: Talled survey results for undergraduate areas of interest

Count	Area
10	AI / Learning Systems
8	Algorithms
7	Theory
7	Graphics
6	Systems
3	Languages
2	Undecided
2	Software Engineering
2	Distributed Systems
1	Vision
1	Computational Biology
1	Networking
1	Information Theory

Table A.8: What areas are overrepresented or underrepresented? Are any of the CS requirements inappropriate or outdated?

Count (Over-/Correctly/Under- represented)	Area
4/0/0	Graphics
1/0/0	Geometry
0/1/1	AI/Learning Systems
0/0/1	“Correctness” classes (CS116/118/119)
0/0/2	Compilers
0/0/2	“Everything”
0/0/2	Languages
0/0/2	“Practical” CS courses
0/0/2	Software Engineering
0/0/3	Algorithms
0/0/3	Systems
0/0/3	Theory

Are you satisfied overall with the organization of courses in the CS curriculum? For example, do you think all of the sequences are structured well? Are there courses which overlap in material too much?

- A project course in theory would be “neat,” and perhaps CS129 would satisfy that if it were offered in full.
- We need more diversity in course offerings.
- We need More consistency in course offerings each year.
- “Offer the courses first. Then I’ll worry about this.”
- CS145 needs to be more undergraduate-friendly.

What are the biggest changes you would like to see in the CS course offerings?

- More diversity in upperlevel courses needs to be offered (algorithms, compilers, languages, systems) in addition to existing areas.
- More project courses need to be offered.
- Change CS1 to a more useful language and have CS2 cover material that CS1 used to

(Numerous responses were received to the effect of offering *more courses*)

What are the biggest changes you would like to see in the CS option requirements?

- CS1 and CS2 seem unnecessary as prerequisites.
- CS101 should satisfy the upperlevel CS requirement.
- CS1 and CS2 placement should be permitted after students arrive here as frosh.

What areas would you like to see new/more CS courses offered in? *Algorithms, theory, AI, systems, compilers, languages, software engineering.*

Do you intend to complete a CS thesis? Why or why not?

- No, I completed a project sequence.
- No, I hadn’t had sufficient course work in an area until recently.
- No, we lack course work in my primary areas of interest (languages).
- No, not interested in research.
- No, professors are not available to sponsor one.
- No, I didn’t realize this was available.
- I don’t have enough information to decide.
- Yes, I want to have a thesis in preparation for graduate school.

The CS department is considering the creation of a one-unit first-term course consisting of weekly research overview talks by CS professors to introduce undergraduates to research in the department. Would this be useful to you, for example in helping identify SURF or thesis opportunities? To what extent does Lunch Bunch already meet these needs?

- Good idea to have a separate course. Lunch bunch feels geared towards another audience, typically a more highly specialized and advanced audience.
- Lunch bunch meets these needs and may be redundant. Interaction with graduate students and faculty might be better at lunch bunch than in this course.
- Seems like a good way of exposing undergraduates to “real” CS

Have you attended a Lunch Bunch seminar?

- 18 - Yes
- 18 - No

Do you have any further opinions or ideas on how the internet could be used by the CS department for courses?

- Online courses are a short-term fix to our course offering problems.
- Teach the courses on a regular basis (every other year) in real life and offer the course materials online in alternate years.
- Replace Moodle with a better interface.
- Have a centralized location for CS course websites on par with the math department.

Do you have any other concerns or thoughts of which you would like to inform the CS SFC committee?

- Improve the CS cluster as it is poorly managed and is of little use to students.
- “Get rid of the CS option for undergraduates until the E&AS division is willing to fully support it.”
- “Don’t let all the complaints weigh you down, and do the best you can.”
- Offer the classes *in the catalog*

Table A.9: How would the addition of the designation “with honors” to a diploma to students who complete a thesis affect your likelihood of completing a CS thesis?

Count	Response
10	No
8	Yes
2	Yes, but “some projects require even more work than a thesis and wouldn’t get this recognition”
2	Slight increase
2	I don’t know

Table A.10: The CS department is considering offering some courses in a new format. These courses would have much of their material on the internet. There would be face-to-face problem solving sessions with a faculty member or instructor each week. As such, the course is not completely online. The department hopes there will be closer interactions among students and faculty in problem-solving sessions than in many lectures. Most lecture material, other than problem-solving sessions, will be online. Students will be expected to read the material before coming to weekly problem-solving sessions. There would probably be one hour of problem-solving sessions per week as opposed to two and a half hours of lecture. How do you feel about this proposal? Please comment on any suggestions or opinions you might have as well as how likely you would to take classes in this new format.

Count	Opinion/Attitude
13	Good idea
7	Opposed (“I’m paying 200k for real courses. Give me real courses.” Will “distance the students further from professors.” “This solution is suboptimal.”)
3	Skeptical of whether it would “work in practice”
2	Short term fix for offering courses (“Preferable to not having the classes offered”)
1	Good idea for some courses, but not useful for others (“theory and interesting material”)
1	Problem-solving sessions are not a complete substitute for lectures
1	Indifferent
1	Record all lectures
1	Need more information to decide

B Historic Course Offerings

This table was compiled by analysis of previous schedules posted on regis.caltech.edu over the 2006-09 academic years. For one- or two-term courses, the terms the course was offered are noted (i.e., SP08-09). For year-long courses, or those offered every term (such as CS11), the span of years is noted (08-09). For multiterm sequences only partially offered—or in progress, such as CS141a in the 2008-09 academic year—such as a single term, the terms offered are noted in parentheses. For courses offered in the same term over successive years, the offering dates are denoted by SP07-09, for instance to indicate the spring term of the 2007-08 and 2008-09 academic years.

CS101 and CS286 are omitted due to their intentionally variable nature.

Number	Title	Offered
CS1	Intro. Computation	FA06-09
CS2	Intro. Programming Methods	WI06-09
CS3	Intro. Software Engineering	SP06-09
Ma/CS6abc (*)	Intro. Discrete Math	06-09
CS11	Computer Language Shop	06-09
CS21	Decidability and Tractability	WI06-09
CS24	Intro. Computing Systems	SP06-09
CS38	Intro. Algorithms	SP06-09
CS42	Intro. Relational Databases	WI08-09
CS47	Advanced Object-Oriented Programming	FA06-07
EE/CS51 (*)	Principles of Microprocessor Systems	WI06-09
EE/CS52 (*)	Microprocessor Systems Laboratory	SP 06-07, WI/SP 07-09
EE/CS53 (*)	Microprocessor Project Laboratory	06-09
EE/CS54 (*)	Advanced Microprocessor Projects Laboratory	06-08, FA/SP08- 09
CS/EE/ ME75abc	Introduction to Multidisciplinary Systems Engineering	06-07
CS116	Reasoning about Program Correctness	FA08-09
Ma/CS117abc (*)	Computability Theory	06-07,08-09
CS118	Logic Model Checking for Formal Software Verification	WI07-09
CS119	Reliable Software: Testing and Monitoring	SP07-09
CS/EE/ Ma129abc (†)	Information and Complexity	06-07 (abc), WI/SP 07-08 (abc), FA08-09 (a)

ME/CS132 (*)	Advanced Robotics: Navigation and Vision	WI06-07, WI08-09
CS134abc (†)	Computing Systems	06-07 (abc), SP08-09 (a)
CS136abc (†)	Programming Languages Laboratory	07-08
CS138a	Computer Algorithms	SP08-09 (a)
CS141a (†)	Distributed Computation Laboratory	07-08 (abc), SP08-09 (a)
CS/EE145abc (†)	Networking	07-09
EE/CNS/CS 148ab (*)	Selected Topics in Computational Vision	FA/SP 06-07 (ab), SP07-08 (b)
CS150	Probability and Algorithms	WI06-08
CS151	Complexity Theory	SP06-07, SP08-09
CS156a	Learning Systems	FA/WI 07-08 (ab), FA08-09 (a)
CS/CNS171 (‡)	Intro. to Computer Graphics Laboratory	FA06-09
CS/CNS174 (‡)	Computer Graphics Projects	SP06-09
CS175 (‡)	Geometric Modeling	SP06-07
CS176 (‡)	Intro. Computer Graphics Research	WI06-09
CS177 (‡)	Discrete Differential Geometry: Theory and Applications	FA06-09
CS/EE181abc († *)	VLSI Design Laboratory	06-07, 08-09
CS 185a	Asynchronous VLSI Design Laboratory	FA07-08
CNS/Bi/Ph/ CS187	Neural Computation	FA06-09
CS/CNS/Bi 191ab	Biomolecular Computation	WI/SP 06- 07
Ph/CS219abc (*)	Quantum Computation	FA/WI06- 07 (abc), 08-09 (abc)
SS/CS 241a	Introduction to Social and Information Sciences	WI06-08

- (*)- Denotes a course offered by an instructor outside of the CS department.
- (†)- Denotes a course which satisfies the project sequence requirement for the undergraduate CS option.
- (‡)- The combination of “CS/CNS 174 and two other CS 170-series courses” satisfy the project sequence requirement.

Courses listed in the catalog, but not offered during the period surveyed:

- ACM/CS 114 ab - Parallel Algorithms for Scientific Applications.

- CS 134bc - Computing Systems
- CS 138bc - Computer Algorithms
- CS 139abc - Concurrency in Computation
- CS/EE 184 ab. Computer Architecture
- CNS/CS/EE 188. Topics in Computation and Biological Systems

C New Course Proposals and Descriptions

- **CS 1 Introduction to Computer Programming.** 9 units (3-4-2); first term. CS 1 is a course on computer programming emphasizing the program design process and pragmatic programming skills. The course will use the Python programming language and will not assume previous programming experience. Material covered will include data types, variables, assignment, control structures, functions, scoping, compound data, string processing, modules, basic input/output (terminal and file), as well as more advanced topics such as recursion, exception handling and object-oriented programming. Program development and maintenance skills including debugging, testing, and documentation will also be taught. Assignments will include problems drawn from fields such as graphics, numerics, networking and games. At the end of the course, students will be ready to learn other programming languages in courses such as CS 11, and will also be ready to take more in-depth courses such as CS 2 and CS 4. Instructor: Vanier.
- **CS 4 Fundamentals of Computer Programming.** 9 units (3-4-2); second term. Prerequisite: CS 1 or equivalent. CS 4 is a course in computer programming which gives students the conceptual background necessary to construct and analyze programs, which includes specifying computations, understanding evaluation models, and using major programming language constructs (functions and procedures, conditionals, recursion and looping, scoping and environments, compound data, side effects, higher-order functions and functional programming, and object-oriented programming). CS 4 emphasizes key issues that arise in programming and in computation in general, including time and space complexity, choice of data representation, and abstraction management. This course is intended for students with some programming background who want a deeper understanding of the conceptual issues involved in computer programming. This understanding is critical in helping programmers write correct, maintainable, extensible, and efficient programs. Instructor: Vanier.
- **EE/CS/Ma 127 Error-Correcting Codes.** 9 units (3-0-6); second, third terms. Prerequisite: Ma 2. This course, which is a sequel to EE/Ma 126 a, but which may be taken independently, will develop from first principles the theory and practical implementation of the most important techniques for combatting errors in digital transmission or storage systems. Topics include algebraic block codes, e.g., Hamming, Golay, Fire, BCH, Reed-Solomon (including a self-contained introduction to the theory of finite fields); convolutional codes; and concatenated coding systems. Emphasis will be placed on the associated encoding and decoding algorithms, and students will be asked to demonstrate their understanding of these algorithms with software projects.
- **ME/CS 131 Advanced Robotics: Manipulation and Sensing.** *Added CS cross-listing to the course number.*

- **CS/EE 143 Communication networks.** 9 units (3-3-3) first term. Prerequisite: Ma 2 ab. This course introduces the basic mechanisms and protocols in communication networks, and mathematical models for their analysis. It covers topics such as digitization, switching, switch design, routing, error control (ARQ), congestion control, layering, queuing models, optimization models, basics of protocols in the Internet, wireless networks, and optical networks. This course can be combined with CS/EE 144 and CS/EE 145 to satisfy project requirement for CS undergraduate degree. Instructor: Low.
- **CS/EE 144 Ideas behind the web.** 9 units (3-0-6) second term. Prerequisite: Ma 2 ab. The web is an essential part of our lives and we all depend on it every day, but do you really know what makes it work? This course studies the “big” ideas behind the web. Things like, how do search engines work? How can search engines make so much money from putting ads next to its search results? Are there ways to prevent spammers from accumulating lots of email addresses? What does the web actually look like? How big is the web? For all these questions and more, the course will provide a mixture of both mathematical models and real-world hands-on labs. This course can be combined with CS/EE 143 and CS/EE 145 to satisfy project requirement for CS undergraduate degree. Instructor: Wierman.
- **CS/EE 145 Projects in networking.** 9 units (0-0-9) third term. Prerequisite: CS/EE 143, CS/EE 144. Students are expected to execute a substantial project in networking, write up a report describing their work, and make a presentation. This course can be combined with CS/EE 143 and CS/EE 144 to satisfy project requirement for CS undergraduate degree. Instructor: Low/Wierman.
- **CS/EE 146 Advanced networking.** 9 units (3-3-3) second term. Prerequisite: CS/EE 143 or instructor’s permission. This is a research-oriented course meant for undergraduates and beginning graduate students who want to learn about current research topics in communication networks. The topics covered in the course will vary, but will be pulled from current research topics in the design, analysis, control, and optimization of networks, protocols, and internet applications. Instructor: Low.
- **CS/EE 147 Network performance analysis.** 9 units (3-0-6) third term. Prerequisite: Ma 2ab is required. CS/EE 143, CS/EE 144, and ACM 116 are recommended. When designing a network protocol, distributed system, etc. it is essential to be able to quantify the performance impacts of design choices along the way. For example, should we invest in more buffer space or a faster processor? One fast disk or multiple slower disks? How should requests be scheduled? What dispatching policy will work best? Ideally, one would like to make these choices before investing the time and money to build a system. This class will teach students how to answer this type of “what if” question by introducing students to analytic performance modeling, the tools necessary for rigorous system design. The course will focus on the mathematical tools of performance analysis (which include stochastic modeling, scheduling theory, and queuing theory) but will also highlight applications of these tools to real systems. Instructor: Wierman.
- **CS/CNS/EE 154 Artificial Intelligence.** 9 units (3-3-3) First term. Prerequisite: Ma 2 ab. How can we build systems that perform well in unknown environments and unforeseen situations? How can we develop systems that exhibit “intelligent” behavior, without prescribing explicit rules? How can we build systems that learn from experience in order to improve their performance? We will study core modeling techniques and algorithms from statistics, optimization, logic planning and control and study applications in areas such as sensor networks, robotics and the internet. The course is designed for upper level undergraduate and

graduate students, and will provide hands-on experience in a course project. Prerequisites are basic knowledge in probability, statistics and algorithms. Instructor: Krause. Not offered in 2009-10

- **CS/CNS/EE 155 Probabilistic Graphical Models.** 9 units (3-3-3) Second term. Prerequisite: CS/EE/CNS 154 or CS/CNS/EE 156a or instructor's permission Many real world problems in AI, computer vision, robotics, computer systems, computational neuroscience, computational biology and natural language processing require to reason about highly uncertain, structured data, and draw global insight from local observations. Probabilistic graphical models allow addressing these challenges in a unified framework. These models generalize approaches such as hidden Markov models and Kalman filters, factor analysis and Markov random fields. In this course, we will study the problem of learning such models from data, performing inference (both exact and approximate) and using these models for making decisions. The techniques draw from statistics, algorithms and discrete and convex optimization. The course will be heavily research oriented, covering current developments such as probabilistic relational models, models for naturally combining logical and probabilistic inference and nonparametric Bayesian methods. The course is designed for graduate students and advanced undergraduate students in CS, EE, ACM, CDS and CNS. Background in algorithms and statistics will be required prerequisites. Instructor: Krause.
- **CS/CNS/EE 156ab Learning Systems.** 9 units (3-3-3) Prerequisites: Ma 2 and CS 2, or equivalent. Introduction to the theory, algorithms, and applications of automated learning. How much information is needed to learn a task, how much computation is involved, and how it can be accomplished. Special emphasis will be given to unifying the different approaches to the subject coming from statistics, function approximation, optimization, pattern recognition, and neural networks. Instructor: Abu-Mostafa.
- **CS/CNS/EE 159 Projects in Machine Learning and AI.** 9 units (0-0-9) Third term. Prerequisite: CS/EE/CNS 154 or CS/CNS/EE 156b Students are expected to execute a substantial project in AI and/or machine learning, write up a report describing their work, and make a presentation. This course can be combined with CS/CNS/EE 154/155 or with CS/CNS/EE 156ab to satisfy the project requirement for CS undergraduate degree. Instructor: Krause. Not offered in 2009-10
- **CS/EE 245 Special topics in networking.** 9 units (3-3-3) second or third term. Prerequisite: CS/EE 143 and CS/EE 144 or instructor's permission. This course is an advanced, research-oriented seminar in communication networks meant for graduate students and advanced undergraduates. The topics covered in the course will vary, but will always come from the cutting-edge of networking research. Examples of possible topics are: error correcting codes, wireless networking, scheduling theory, and congestion control. Instructor: Staff. Not offered 2009-10.
- **CS/CNS/EE 255 Special Topics in Machine Learning.** 9 units(3-3-3) This course is an advanced, research-oriented seminar in machine learning and AI meant for graduate students and advanced undergraduates. The topics covered in the course will vary, but will always come from the cutting-edge of machine learning and AI research. Examples of possible topics are: active learning and optimized information gathering, AI in distributed systems, computational learning theory, machine learning applications (on the web, in sensor networks and robotics). Instructor: Krause

- **Current Topics in Theoretical Computer Science.** 9 units (3-0-6). Third term. Prerequisites: CS21 and CS38, or permission of the instructor. May be repeated for credit, with permission of the instructor. Students in this course will study an area of current interest in theoretical computer science. The lectures will cover relevant background material at an advanced level, and present results from selected recent papers within that year's chosen theme. Students will be expected to read and present a research paper. To be offered in alternate years. Instructors: Schulman, Umans.