University of Illinois Department of Computer Science
Goals, Objectives, and Assessments
January 2009

Introduction

The University of Illinois Computer Science Department is among the oldest CS departments in existence. It was among the first to offer a Bachelor of Science degree in Computer Science, granting its first BS degree in 1973. It remains among the most distinguished in the nation, routinely ranking in the top ten in surveys of overall quality. We view our role as one of training students to work at the forefront of computer science, whether in the industrial world or in academia.

Our curriculum is marked by an emphasis on fundamentals: basic science and mathematics, computer architecture, theory of computation, and algorithm development and analysis. Although our primary emphasis is on the construction of software, our emphasis on fundamentals makes our students well-rounded software engineers and computer scientists, allows them to understand new developments in the computer field, and, above all, gives them the tools to be lifelong learners.

Specific goals for our graduates are contained in our Mission Statement :

"The goal of our undergraduate program is […] educating our students in the theory and application of computer technology as well as any college or university in the world.

"We believe that when our students graduate, they should possess:

- A thorough understanding of computer technology
- A high level of practical skill in the use of that technology
- The ability to conceive, design, and implement software systems, using appropriate existing technologies where available, and producing creative solutions when necessary
- The ability to work in teams, with other computer scientists as well as non-computer scientists; to communicate, orally and in writing, with specialists and non-specialists, about computer technology
- Integrity in their professional dealings, and sensitivity to ethical issues that arise in society's use of computers
- Detailed knowledge of at least one advanced area of Computer Science
- For students who contemplate a career in research, experience working in a research group, and knowledge of the mores and ethics of the research community

"Above all, our students must possess the intellectual tools that will enable them to keep pace with the ever-changing world of Computer Science."

Implementation of Objectives

Objective	How measured	When measured	Improvements Identified	Improvements Implemented
Thorough understanding of computer technology	Exams, lab work, and written homework in CS classes	Continually	This is an area in which all survey instruments show that the CS department is doing a good job.	
High level of practical skill in the use of computer technology	Feedback from upper-level course instructors	Continually	Students have insufficient programming skills coming out of lower-level courses	Curricular changes include addition of two required courses: lower-level systems programming course and "programming studio"
	Feedback from employers	Continually	Students need better programming skills, especially in low-level programming	Curricular changes include addition of "programming studio," systems programming course, and more programming in the second computer architecture course
The ability to conceive, design, and implement software systems	*Same as previous item*			
Ability to work in teams	Engineering College Alumni surveys	Every seven years	Students should have more experience	Curriculum revision added required courses with group projects.
	Coursework in Senior Project I and II, Software Engineering I and II, and other upper-level courses	Continually	Students needed more practice in working in teams	Senior Project or Software Engineering now required of all students (except those doing senior thesis)
	Feedback from employers	Continually	Students have insufficient skills in	Curricular changes include "programming studio" and new requirement for one-

			teamwork	year senior project course
Communicate clearly orally	ABET accreditation	2001	Lack of practice in oral communication cited as a concern in last ABET review	Added CS 210, Ethical and Professional Issues in CS, for which the main assignment is an oral presentation
	Feedback from employers	Continually	Students needed more practice in public speaking, especially on technical subjects	New required course (CS 242, Programming Studio) involving weekly technical presentations
	CS Undergraduate Town Hall	2007	Students have expressed the need for more work on this in courses	Looking into adding oral presentation requirement to senior capstone courses
Communicate clearly in writing	Comp I and Advanced Comp requirements	Continually		Senior Project, Software Engineering, or Senior thesis now required of all students; all entail significant technical writing
Integrity in professional dealings, and sensitivity to ethical issues that arise in society's use of computers	ABET accreditation	2001	Lack of coursework material covering ethics and contemporary issues cited as a concern	Instituted requirement that all CS majors complete CS 210 Ethical and Prof'l Issues in CS in 2002
Detailed knowledge of at least one advanced area of Computer Science	Measured in courses		Curriculum changes made to include area of concentration in CS;	
Research experience for students interested in pursuing careers in research	Graduate admissions committee	Annual	Need to have more students getting research experience	1. New curriculum includes senior thesis as one of required options in senior year. 2. Honors students now required to do senior thesis 3. New web page

				describing undergraduate research possibilities.
Lifelong learners possessing intellectual tools to keep pace with the ever-changing world of Computer Science	Engineering College Alumni surveys	Every seven years	None	None
	CS Department Executive Advisory Council	Continually	None	None
	Director of Alumni Relations	Continually	None	None

Assessments

Informal assessments

Our overall educational outcomes are measured, in the final analysis, through the eyes of our "consumers:" employers, faculty who advise our graduates in graduate school, and our alumni themselves.

Employers express nearly unanimous approbation of our students. Our students are recruited by virtually every major computer-oriented company. Our alumni include numerous prominent members of the computer science community.

Students continually provide feedback to instructors and the departmental academic office. The web page http://www.cs.uiuc.edu/undergraduate/feedback.php, linked prominently on our departmental web page, gives clear directions on where and how to ask for guidance, lodge complaints, and make suggestions.

The Computer Science Student Advisory Committee (CSSAC) is another avenue for informal assessment of the program and its courses. CSSAC allows student representatives to address problems in the courses and curriculum by meeting with representatives of the department's faculty and staff and devising mutually arrived at solutions. One of the activities of CSSAC is the annual CS Undergraduate Town Hall Meeting, where any student in the department may raise issues that they feel the department should address.

Including student representatives in key departmental committees – most notably the undergraduate studies committee – allows students to provide input on courses and curricula matters and on departmental policies.

The CS Executive Advisory Council (EAC), our alumni board, meets twice a year in October and April. Board members include numerous industry leaders and generous supporters of the department from graduating classes spanning several decades. The board advises the director of development and alumni relations and the department of computer science leadership and professional staff on matters central to the department's educational and research missions: the recruiting of students, outreach efforts to women and minorities, corporate relations, labor issues, and many other development and advancement issues. Another informal assessment is the student-run "CourseGuide." Created by the CS student club !Bang, the CourseGuide offers comparative statistics on the overall quality and difficulty of courses.

Formal assessments

1. Student course evaluations. All faculty must allow classroom time toward the end of every semester for students to complete a standard questionnaire (the Instructor and Course Evaluation System, or ICES, form). The results are returned to the faculty member after the end of the semester; numerical ratings in twenty areas, as well as written comments, are used for self-assessment and improvement. Summary numerical data are given to the department head, who uses them as input in annual faculty evaluations.
2. The department recently added an online mid-semester evaluation for all CS courses.
3. The University conducts a survey of seniors each year.
4. The department has administered end-of-semester and end-of-year surveys at the end of the Fall 2004, Fall 2006, Spring 2006 and Spring 2007 semesters. These surveys assess course management and advising issues (Fall) and provide information on student placement in internships, jobs and/or graduate programs (Spring).
5. In April 2007, the University of Illinois Center for Teaching Excellence conducted a focus group of graduating seniors to assess their satisfaction with the program.
6. The college Career Services Office publishes an annual report on student placements.
7. Every five years, the College of Engineering conducts surveys of its graduates five years after graduation – the last time in 2003. We now have data from the graduating class of 1998.

D. Program Improvement

Major changes within the last five years:

- CS 210 – Ethical and Professional Issues in CS introduced in 2002. Addressed a concern highlighted in previous ABET accreditation cycle (2001).
- Curriculum revision, effective Fall 2005, added these new requirements:
    - CS 241 – System Programming. Introduced to allow students necessary preparation for taking upper-level systems courses earlier in their college career.
    - CS 242 – Programming Studio. Introduced to address concerns that many students' programming and presentation skills were inadequate for upper-level courses.
    - Capstone senior project/thesis, Software Engineering course or senior thesis now required; addresses need for more experience in teamwork, and in written and oral presentation.
- Opening of the Thomas M. Siebel Center for Computer Science, the department's new home.
- Substantial decrease in student-to-faculty ratio (down from around 30/1 to 15/1)
- Substantial decrease in student-to-advisor ratio (20/1 down from around 70/1)
- Hiring of a full-time academic advisor specifically for undergraduates
- Decrease in size of 400-level courses (87.9 students per course in Fall 2002 versus 49.5 students per course in Fall 2006 – excluding special topics courses), and assignment of multiple instructors to several courses (especially CS 433 Computer

System Organization, CS 473 Algorithms, every spring semester).

- Added a suite of Computer Security courses: CS 461 Intro to Information Assurance, CS 463 Computer Security and CS 460 Security Laboratory
- Added a course in Human-Computer Interfaces: CS 465 – Principles of User Interface Design, approved August 2004
- Revised the theory sequence, effective Fall 2005: The previous sequence was a total of 8 hours, and was strong on discrete math and algorithms, but short-changed theory of computation (machine models, computability, etc.). The new sequence is 9 hours, with a full 3-hour course devoted to each of these three areas
- Advanced Comp credit approved for Senior Project II, Software Engineering II – encourages students to work on field-specific written communication
- Freshman orientation programs to help develop early sense of community
    - College of Engineering instituted two-day, residential orientation program for women entering the college
    - CS department instituted orientation and activities for all new CS students

Significant future program improvement plans based upon recent assessments

- Continued decrease in size of 400-level courses
- Adding discussion sections (without adding new material or assignments) to theory courses, thus providing more structured guidance on some of the most difficult courses in the curriculum.
- Adding discussion section to CS 418 – Computer Graphics without adding new material or assignments. Provides students with guidance in implementation issues not covered in lectures (a response to instructor and student feedback: !CourseGuide had indicated that CS 418 is the most time-consuming course in the program.)
- Adding discussion sections to CS 241 – System Programming and adding an hour of credit (a response to student and instructor feedback).
- Numerical Analysis revisions (separate all but most advanced undergrads from grad-level courses)
- Bioinformatics course (CS 466 proposed) – part of bioinformatics initiative