

ABET
Self-Study Report
for the
Bachelor of Science in Computer Engineering
at
University of Washington
Seattle, Washington

June 24, 2013

Table of Contents

Contents

Table of Contents.....	2
BACKGROUND INFORMATION	7
A. Contact Information	7
B. Program History	7
C. Options	7
D. Organizational Structure	8
Figure 0-1 CSE Department Organizational Chart.....	8
E. Program Delivery Modes	9
F. Program Locations.....	10
G. Deficiencies, Weaknesses or Concerns from Previous Evaluation(s) and the Actions Taken to Address Them.....	10
H. Joint Accreditation	11
CRITERION 1. STUDENTS.....	12
A. Student Admissions	12
B. Evaluating Student Performance.....	14
C. Transfer Students and Transfer Courses	15
C.1 Acceptance of Transfer Students	15
C.2 Evaluation of Transfer Credit.....	15
D. Advising and Career Guidance	16
D.1 Advising.....	16
D.2 Career Guidance.....	16
E. Work in Lieu of Courses	17
F. Graduation Requirements	18
CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES	19
A. Mission Statement.....	19
B. Program Educational Objectives.....	19
C. Consistency of the Program Educational Objectives with the Mission of the Institution ...	19
D. Program Constituencies	20
E. Process for Revision of the Program Educational Objectives.....	20
CRITERION 3. STUDENT OUTCOMES	22
A. Student Outcomes	22
B. Relationship of Student Outcomes to Program Educational Objectives.....	22
CRITERION 4. CONTINUOUS IMPROVEMENT.....	24
A. Student Outcomes	24
A.1 Data Sources	24
A.2 Assessment Process Overview.....	26
A.3 Targeted Assessment of Outcomes in Courses	28
A.4 Additional Assessment Processes	33

B. Continuous Improvement.....	42
B.1 Targeted Assessment of Outcomes in Courses	42
B.2 Additional Assessment Processes	45
CRITERION 5. CURRICULUM	49
A. Program Curriculum.....	52
A.1 Tables.....	52
A.2 Curriculum Alignment with Program Educational Objectives	58
A.3 Curriculum & Student Outcomes.....	59
A.4 Course Prerequisite Structure	62
A.5 Meeting Degree Requirements	63
A.6 Design Experience	66
A.7 Extra-Curricular Activities and Co-op.....	77
A.8 Materials Available to Committee	77
B. Course Syllabi	77
CRITERION 6. FACULTY.....	78
A. Faculty Qualifications	78
Table 6-1 Faculty Qualifications	80
B. Faculty Workload.....	85
Table 6-2 Faculty Workload Summary	85
C. Faculty Size	90
D. Professional Development	91
E. Authority and Responsibility of Faculty	91
CRITERION 7. FACILITIES.....	92
A. Offices, Classrooms and Laboratories	92
B. Computing Resources	95
C. Guidance	95
D. Maintenance and Upgrading of Facilities	95
E. Library Services	96
F. Overall Comments on Facilities	96
CRITERION 8. INSTITUTIONAL SUPPORT	97
A. Leadership.....	97
B. Program Budget and Financial Support	97
C. Staffing.....	98
D. Faculty Hiring and Retention.....	98
E. Support of Faculty Professional Development.....	100
PROGRAM CRITERIA	104
Appendix A – Course Syllabi	107
Written & Oral Communication	107
HCDE 231 – Introduction to Technical Writing	108
Mathematics & Natural Sciences.....	110
CHEM 142 General Chemistry	111
CHEM 152 General Chemistry.....	113
CHEM 162 General Chemistry.....	115
MATH 124 Calculus with Analytic Geometry I	116

MATH 125 Calculus with Analytic Geometry II	118
MATH 126 Calculus with Analytic Geometry III.....	119
MATH 307 Introduction to Differential Equations	120
MATH 308 Matrix Algebra with Applications	121
MATH 309 Linear Analysis	122
PHYS 121 Mechanics.....	123
PHYS 122 Electromagnetism	124
PHYS 123: Waves	125
STAT/MATH 390: Probability and Statistics in Engineering and Science.....	127
STAT 391: Probability and Statistics for Computer Science	128
Required.....	129
CSE 142 Computer Programming I.....	130
CSE 143 Computer Programming II.....	132
CSE 311 Foundations of Computing I.....	134
CSE 312 Foundations of Computing II	136
CSE 332 Data Abstractions	138
CSE 351 The Hardware/Software Interface	140
CSE 352 Hardware Design and Implementation.....	142
EE 205 Introduction to Signal Conditioning	144
EE 215 Fundamentals of Electrical Engineering.....	146
CE Senior Electives	148
CSE 331 Software Design and Implementation	149
CSE 333 Systems Programming.....	151
CSE 341 Programming Languages.....	153
CSE 344 Introduction to Data Management.....	155
CSE 401 Introduction to Compiler Construction	157
CSE 403 Software Engineering	159
CSE 421 Introduction to Algorithms	161
CSE 427 Computational Biology	163
CSE 428 Computational Biology Capstone.....	165
CSE 431 Introduction to Theory of Computation	167
CSE 440 Introduction to HCI: User Interface Design, Prototyping, and Evaluation	168
CSE 441 Advanced HCI: Advanced User Interface Design, Prototyping, and Evaluation	170
CSE 444 Database Systems Internals	172
CSE 446 Machine Learning.....	174
CSE 451 Introduction to Operating Systems	176
CSE 452 Introduction to Distributed Systems.....	178
CSE 454 Advanced Internet and Web Services.....	179
CSE 455 Computer Vision	181
CSE 456 Story Design for Computer Animation	183
CSE 457 Computer Graphics.....	185
CSE 458 Computer Animation.....	187
CSE 459 Pre-Production for Collaborative Animation	188
CSE 460 Animation Capstone	190
CSE 461 Introduction to Computer-Communication Networks	192
CSE 464 Advanced Topics in Digital Animation.....	194

CSE 466 Software for Embedded Systems.....	196
CSE 467 Advanced Digital Design	198
CSE 471 Computer Design and Organization	200
CSE 472 Introduction to Computational Linguistics.....	202
CSE 473 Introduction to Artificial Intelligence.....	204
CSE 477 Digital System Design.....	206
CSE 481A Capstone Software Design: Operating Systems	208
CSE 481B Capstone Software Design: Tablet PC.....	210
CSE 481C Capstone Software Design: Robotics	212
CSE 481D Capstone Software Design: Games	214
CSE 481H Capstone Software Design: Accessibility	216
CSE 481I Capstone Software Design: Sound.....	218
CSE 481K Capstone Software Design: Resource-constrained Environments	220
CSE 481O Capstone Software Design: 3D Cameras	222
CSE 484 Computer Security.....	224
CSE 486 Introduction to Synthetic Biology	226
CSE 487 Advanced Systems and Synthetic Biology.....	228
CSE 488 Laboratory Methods in Synthetic Biology	230
CSE 490 Patent Law for Engineers	232
CSE 498 Senior Project	234
Appendix B – Faculty Vitae	236
Richard Anderson	237
Ruth E. Anderson.....	239
Thomas E. Anderson.....	241
Magdalena Balazinska	243
Paul Beame	245
Gaetano Borriello.....	247
Luis Ceze	249
Brian L. Curless	251
Pedro Domingos	253
Oren Etzioni.....	255
Ali Farhadi	257
James Fogarty	259
Dieter Fox	261
Shyamnath Gollakota.....	263
Dan Grossman.....	266
Bruce Hemingway	268
Anna R Karlin.....	270
Ira (Irena) Kemelmacher-Shlizerman	272
Tadayoshi Kohno	274
Arvind Krishnamurthy.....	276
Edward D. Lazowska.....	278
James R. Lee.....	280
Henry M. Levy.....	282
Hélène Martin	284
Mausam.....	285

Barbara Mones	287
Mark Oskin	289
Shwetak Patel.....	291
Hal Perkins.....	293
Anup Rao	294
Walter L. Ruzzo	295
Steve Seitz.....	297
Linda G. Shapiro.....	299
Joshua R. Smith	301
Marty Stepp.....	303
Dan Suciu.....	305
Steven L. Tanimoto.....	307
Ben Taskar	309
Emanuil Todorov	311
Martin Tompa	313
Daniel S. Weld.....	314
David Wetherall.....	316
John Zahorjan	319
Luke Zettlemoyer.....	321
Appendix C – Equipment	323
Student Computing Resources (physically accessible)	323
Instructional Staff Computing	323
Instructional Shared Facilities Hosted in Datacenter.....	323
Animation Renderfarm	324
Appendix D – Institutional Summary	325
1. The Institution.....	325
2. Type of Control.....	325
3.. Educational Unit	325
3. 1. University of Washington Engineering Initiative.....	329
4. Academic Support Units.....	334
5. Non-academic Support Units.....	335
6. Credit Unit	336
7. Tables.....	336
Table D-1. Program Enrollment and Degree Data	336
Table D-2 Personnel	338
Signature Attesting to Compliance	339

BACKGROUND INFORMATION

A. Contact Information

John Zahorjan – Professor
Computer Science & Engineering
University of Washington
AC101 Paul G. Allen Center, Box 352350
185 Stevens Way
Seattle WA 98195-2350

Email: zahorjan@cs.washington.edu
Phone: 206 543-0101

B. Program History

The University of Washington's Department of Computer Science & Engineering began as an inter-college graduate program in 1967. In 1975 a Bachelor's program in Computer Science was initiated, targeted to graduate 40 students per year. Departmental status was conferred, and the department was placed under the College of Arts & Sciences. In 1989 the department moved to the College of Engineering, changed its name to the Department of Computer Science & Engineering, and initiated a second Bachelors program -- an ABET-accredited Computer Engineering program targeted to graduate 40 students per year. In 1999 the department expanded its Bachelors program in Computer Engineering to 80 graduates per year (160 total Bachelors graduates per year, equally divided between Computer Science and Computer Engineering). In 2012, we increased total program enrollment to 200 students per year.

As of 2013, the Department of Computer Science and Engineering continues to offer a **Bachelor of Science in Computer Engineering** degree through the College of Engineering and a **Bachelor of Science in Computer Science** degree through the College of Arts & Sciences. We do not set quotas between Computer Science and Computer Engineering, so the fraction in each program reflects student demand. Approximately $\frac{1}{3}$ of our students pursue the Computer Engineering degree and $\frac{2}{3}$ Computer Science. Although our students are differentiated by the degree program they choose, they are otherwise treated very similarly with respect to admission, advising, and in the classroom.

Our last ABET review was in 2007. From 2008-2011 we went through a curriculum update. The changes we made are described in detail in Criterion 5 of this document. Summarizing, the changes include:

- Removed old 300 level courses and introduced new 300 level courses.
- Restructured program to remove specializations and make the degree more cohesive.
- Added specific math requirement to ensure all students were completing 45 required credits of math and science.

C. Options

Prior to Autumn 2011, we offered two specializations, one focused on hardware and one focused on software. We now have a single set of requirements, and rather than thinking of students as headed in one of two broad directions, we have enabled a wider variety of more focused specializations.

To help students with planning, we established guidelines for pursuing various "pathways" within the major. A pathway is a set of courses that we suggest is most useful for a student whose career goals fall in a certain direction. The pathways were established by collecting feedback from industry representatives and faculty. The suggestions are made available to students through a web page, <http://www.cs.washington.edu/students/ugrad/pathways/>, as well as through more direct-contact avenues. The current list of pathways is Databases; Data Mining; Embedded Systems; Graphics, Vision, Games and Animation; Hardware Design/Engineering; Information and Communications Technologies for Development (ICTD); Interactions with Biology, Bioengineering, Medicine and Genomics; Mathematical Foundations; Ph.D. in CSE; Robotics and Control Systems; Scientific Computing; Security; Systems Development; Software Engineering; User Interface and Human-Computer Interaction; and Web Development.

D. Organizational Structure

The department leadership consists of the Chairman, two Associate Chairs (one focusing on educational activities and the other focusing on development and external relations), and an Executive Committee consisting of these three individuals plus four elected members.

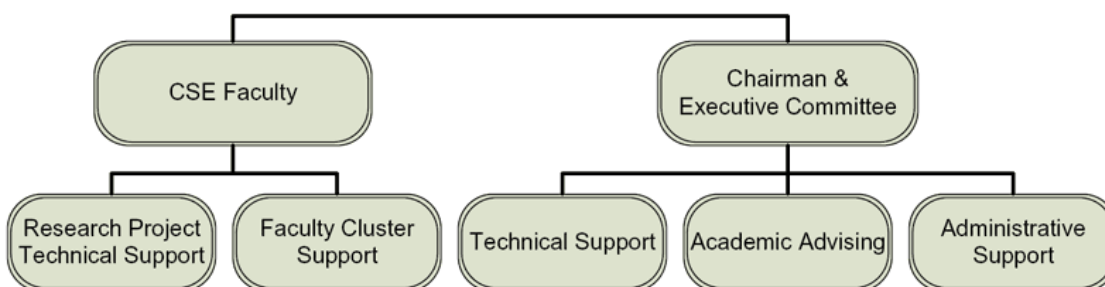


Figure 0-1 CSE Department Organizational Chart

The Chairman is appointed by the Dean for a five-year term. Hank Levy was appointed in 2006 and recently agreed to a three-year extension, which ends in September 2014. The Associate Chairs are appointed by the Chairman for short terms (one or more years) and for duties that correspond with current departmental needs and individual interests. Additional members of the Executive Committee are elected for one-year terms, with a maximum of two consecutive terms.

The general duties of the Executive Committee are to be “in the loop” on all issues, to deal with straightforward issues without engaging the department as a whole, to approve simple appointments (e.g., Affiliate Faculty and postdocs), to ensure that the department as a whole is engaged on important issues, and to serve as a two-way communication channel.

The department has about 46 tenure-line faculty members, 6 instructors, and around 35 postdocs. We have roughly 30 administrative staff, 15 technical staff members, and 6 educational advisors. The staff is divided into four “clusters”: technical support, academic advising, central administrative, and faculty support. There are another 30 technical/research staff associated with centers supported by the department.

We formed our first external advisory committee in 1999, but the committee had not met in many years. In 2008 we formed a new committee, consisting of Barbara Liskov (MIT, chair), Randy Bryant (CMU), Deborah Estrin (UCLA/Cornell), Tom Mitchel (CMU), Christos Papadimitriou (Berkeley), Pat Hanrahan (Stanford), Eric Horvitz (Microsoft Research), and Jeff Dean (Google). That committee is an important input on many department decisions, including the recent curriculum revision.

E. Program Delivery Modes

Almost all undergraduate students in the department are enrolled full-time on-campus, although some students do attend part-time. Courses are offered during day-time hours all four quarters of the academic year, although many fewer courses are offered in summer. Students must meet a satisfactory progress criterion that requires them to complete three courses towards graduation every quarter they are registered (except summer). Exceptions are granted for cooperative work experiences and extenuating circumstances like medical leaves, and for students nearing graduation with minimal outstanding courses to complete.

Most students enter our program in their sophomore year or even the start of their junior year. This is motivated by state legislative mandates to guarantee articulation of transfer students from the state’s Community Colleges and to ensure students successfully complete prerequisites prior to matriculation in the department. However, it has the negative effect of pushing all the requirements for our program into approximately 6-8 quarters rather than 12, the typical number attended during four years.

Partially in response to the restrictions those mandates place on our curriculum, and partially as a means of recruiting the strongest and most diverse student population possible, we implemented a program that was new to us and to our university in 1999. We offer "direct admission" into the major for a small set of high-achieving high school applicants to the University. By removing the uncertainty about admission to the department, we hope to increase the competitiveness of the University of Washington in attracting these students. Further details on this program are given under Criterion 1A, but the short summary is that this is working extremely well for us and for the University. About a quarter of our majors come through our Direct Admission program and we are considering expanding this program in the future.

Finally, we believe in taking a holistic view to our educational mission. Today’s K-12 students are tomorrow’s UW students. We have an extensive K-12 outreach program called DawgBytes (the UW mascot is the Husky). During 2011-12 over 120 teachers and 1,000 K-12 students were involved in this program, which included two professional development opportunities for teachers, 3 weeks of computer science summer camps for middle and high school girls, and the

hosting of the NCWIT Award for Aspirations in Computing. Further information is available at <http://www.cs.washington.edu/outreach/k12/>.

F. Program Locations

Our program takes place on the University of Washington - Seattle campus. Our department's home is the Paul G. Allen Center for Computer Science & Engineering. Rooms for our courses tend to be near the Paul G. Allen Center, however they are sometimes spread throughout the UW campus.

We also offer three international direct exchange agreements:

- KTH in Stockholm, Sweden
- ETH in Zurich, Switzerland
- Saarland University in Saarbrücken, Germany

All three year-long programs give students the opportunity to take computer science and engineering courses at outstanding international universities with the high likelihood that the classes will transfer back as direct major equivalents. Exchange classes are reviewed thoroughly by lead faculty on each topic in order to ensure that they cover all necessary student outcomes.

The exchange program at KTH in Stockholm, Sweden began in Autumn 2001, and undergraduate students have participated in the program since then. The exchange agreement approves sending up to two CSE undergraduate students and five CSE graduate students to KTH in a given year. Undergraduates tend to have greater interest in the exchange, since graduate student participation depends on individual faculty members having specific research links.

The CSE department piloted the ETH exchange program in Spring 2007, and it continues to thrive. At this time, the program supports sending one or two undergraduates each year. The CSE department piloted the Saarland exchange program in Germany during the 2010-2011 academic year. We hope to continue the program depending on future student interest. At this time, the program can support sending one undergraduate student a year.

G. Deficiencies, Weaknesses or Concerns from Previous Evaluation(s) and the Actions Taken to Address Them

The only concern expressed from our ABET 2007 review was that students might be able to graduate without completing the required 45 credits of math and science. To remedy this possibility, we have now added a component to the degree that requires all students to complete 45 specified credits of math and science. This is checked automatically by the university's Degree Audit Reporting System (DARS). Generally, the 45 credits include the introductory calculus series (15), linear algebra (3), 20 credits of lab science and an additional 3 credits from the approved natural science courses on the CSE website, STAT 390, 391, 394, MATH 307, 309, 334, 335, or AMATH 351, 353. (STAT 391, Probability and Statistics for Computer Science, is recommended.) Math 324, multivariable calculus will be added to this list by winter 2014. The last four credits are covered by CSE 311 and 312 where 2 credits from each apply to math.

H. Joint Accreditation

Our program is not jointly accredited.

GENERAL CRITERIA

CRITERION 1. STUDENTS

A. Student Admissions

It is traditional for UW students to apply to majors at the end of their sophomore year – a situation encouraged by Washington’s higher education policy that anticipates a large number of community college transfers. In recent years, we have led an effort to change this significantly by ensuring direct entry to the major (“direct admits”) for top UW admits – an approach now being adopted by other COE programs to encourage many of Washington’s promising students to choose UW. This is particularly important; with the additional certainty and time in the major this program affords, students can better plan their major requirements and are more likely to consider and undertake research and cooperative education experiences that enhance their education.

Our Undergraduate Advising Office coordinates the departmental undergraduate admissions effort. Admission to the department is highly competitive. As mentioned above, there are two paths into the major (direct and regular admission). Both paths use the same criteria for admission: grades in prior courses, especially in math, science and engineering, a written personal statement, and potential to contribute to the field of computing. In the regular process, students apply to the program after completing a set of program prerequisites. Prerequisite courses include one year of calculus (Math 124, 125, 126 or the honors series 134, 135, 136), two quarters of programming (CSE 142 and 143), 10 credits of science (Physics 121 and 122) and 5 credits of English Composition. An Admissions Committee, comprised of four faculty members, three undergraduate advisors, and a graduate student representative evaluate the applications. Three readers, including two faculty members and one advisor, read each application prior to the committee meeting. The committee makes all final decisions by consensus.

All departmental admissions forms are online and students can complete the process remotely. Once the online application is completed, students receive email confirmation of their submitted application. They receive admission notification from departments about three weeks after the application deadline. There are two standard admissions cycles per year, with deadlines of July 1 for Autumn and Feb 1 for Spring.

Our department is very active in recruiting the best students to our program. We host information sessions twice each quarter during the academic year in order to introduce prospective students to our department. We also present at several targeted sessions across campus for students in the Early Entrance program, Honors program, and various other diverse organizations.

In addition to general recruitment, we are actively trying to increase diversity by more broadly attracting applicants through different mechanisms. In 2004 we began offering two seminars

geared toward increasing female engagement in computing careers. In 2010, we joined the inaugural Pacesetters program with the National Center for Women in Information Technology (NCWIT). Through this program we focus on setting concrete goals with regard to increasing the number of women in our department. After just 2.5 years in the program, our admissions cohort went from 19% women in 2009 to 25% women in 2012. Additionally, we typically send 15 to 20 women students (counting both undergraduate and graduate) to the Grace Hopper Celebration of Women in Computing Conference each year.

In 2006, we created a set of videos to help shape the image of CSE in the broader community. These can be found at: <http://www.cs.washington.edu/WhyCSE>. In 2007, one of our videos, “A Day in the Life” was nominated for a Northwest Regional Emmy Award. In 2011, our lead advisor was the executive producer on a series of College of Engineering videos titled UW Engineers Making a Difference (http://www.engr.washington.edu/curr_students/explore/making_diff_videos.html).

Two years ago we received a large grant that allowed us to hire H  l  ne Martin as a lecturer. H  l  ne is a graduate of our program who taught computer science at a local high school for two years after graduation. H  l  ne coordinates our DawgBytes outreach efforts and was recently nominated by her students for one of the top UW teaching awards.

Our Outreach Team includes our introductory programming instructors, H  l  ne Martin and Stuart Reges (a UW teaching award winner), the advising staff and various other faculty. This team greatly expanded our outreach efforts to include:

- middle school and high school summer camps for girls and boys
- CSTA teacher program hosting
- CSE programming competitions for high school students
- Computer Education week celebration activities
- Inspirational Teacher Banquets every two years
- CS4HS workshops every summer for local teachers of math/science
- CSE 190 seminars for students to partner with local secondary schools, bringing CSE into their classrooms

The impact of these efforts is conveyed in Figure 1-1, which shows the fraction of female students in our two pre-major courses (introductory programming). The overall percent of women enrolled in our undergraduate program is now around 25%.

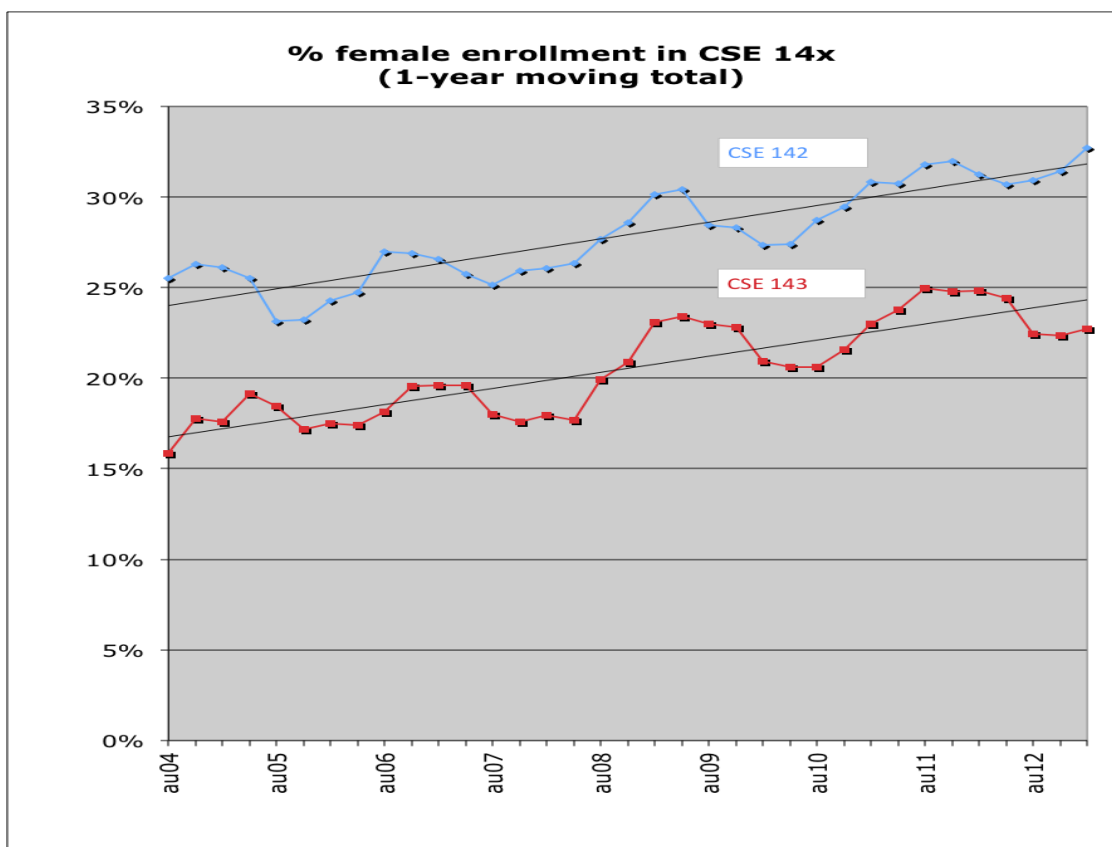


Figure 1-1 – Female enrollment over time in CSE 142 and 143.

B. Evaluating Student Performance

Students are assigned a decimal grade between 0.0 and 4.0 for each course they complete at UW. Instructors, in cooperation with their teaching assistants, are responsible for grades. There are no formal guidelines for the assignment of grades and most of our faculty do not grade on a curve. Our average course grade for undergraduates is 3.2-3.4 depending on the course. This may seem inflationary but is quite consistent with the high GPAs of our incoming students (approx. 3.6). A minimum grade of 2.0 is required for a CSE course to be used toward the program's graduation requirements. Students who do not meet the minimum grade must repeat the course.

Our satisfactory progress policy requires that students must successfully complete three courses toward graduation every quarter. The advising staff reviews student progress every quarter and either notifies students of a probationary quarter for not meeting requirements, or checks in with the student if they are faltering but meeting the minimum requirements. This allows the advising staff to connect struggling students with tutoring and other campus resources that might help them get back on track. It is also routine practice for instructors to consult with advising if they notice a student having unusual trouble during a quarter. Sometimes the advisors have additional context beyond that one course that allows the department to better support the student and offer appropriate help.

The University's registration processes automatically check that course prerequisites are met before registration is allowed. Exceptional cases occur when a student claims to have preparation equivalent to the required prerequisite courses. These requests are typically based on course/courses the student took at another institution, although occasionally they are based on prior work experience. Requests for prerequisite waivers are reviewed by course faculty who evaluate the materials submitted by the student and either grant or deny the waiver. Evaluations are kept in the student's online file, in the notes section of the undergraduate program database.

C. Transfer Students and Transfer Courses

C.1 Acceptance of Transfer Students

Transfer students first apply to and are admitted to the University of Washington, and then they apply separately to the Department through the regular departmental admissions process. There is significant interest by transfer students in our major, and our advising staff meets with many of them during their visit to the UW campus on Transfer Thursdays.

Under the state Higher Education Coordinator Board agreement, students may transfer a maximum of 90 credits towards their degree. Students from Washington State community colleges may petition to have 135 transfer credits applied to their degree. Please note, however, that these courses are all outside the 300 and 400 level CSE courses required for the Computer Engineering Degree. In all cases, the final 45 credits of the degree program must be earned while in residence at the University of Washington.

While technically a student from another four year institution could petition for upper division courses to transfer, community college coursework generally only covers our introductory programming courses (CSE 142 and 143). If courses completed are on our list of pre-approved transfer courses (primarily, from the state's Community Colleges) then credit is granted immediately. If not, then the student is asked to go through the petition process described below in the Evaluation of Transfer Credit section. The possible outcome of a petition is that transfer credit is granted, denied (and the student must take the course in our department for credit), or granted under the condition that the student makes up for some gaps in material (this can take many forms including, but not limited to, an informal conversation with the faculty member evaluating the request, direct study to make up for slight differences to our courses, or the completion of previous course exams and/or assignments). This highly individualized approach is practical only because it is rare. The overwhelming majority of our transfer credits are from Washington State Community Colleges with which we have course articulation agreements. Most other transfer credits are from four-year institutions – usually out-of-state. Courses in engineering that come from an EAC/ABET accredited program are usually accepted readily. Other courses, including from international institutions, are considered with great care.

C.2 Evaluation of Transfer Credit

Transfer credit is accepted subject to the evaluation of each course. The UW provides a centralized evaluation process that we use for non-computing courses. For each course in the major, a petition is submitted online that includes copies of a syllabus, homework assignments, exams and the name of text/s used. There are faculty leads for every course who review the

petitions and accompanying documents and make the final determinations on whether transfer credit should be granted. The two courses that most often fall into this category are our introductory programming courses. The department has a program for the State's community colleges to have their courses pre-approved. We work closely with community college instructors at annual workshops to ensure their introductory programming courses are well-aligned with ours.

D. Advising and Career Guidance

D.1 Advising

The department operates an Undergraduate Advising Office with four full-time professional advisors (Crystal Eney, lead advisor; Raven Alexander, advisor; Elise Dorrough, advisor; Jenifer Pesicka, advisor), a course coordinator (Pim Lustig) and a career point person (David Rispoli, the Professional Master's Program Advisor). The Undergraduate Program Coordinator (Gaetano Borriello) supervises the overall advising effort. The office provides both pre-major advising to prospective students and advising and curriculum planning to current majors. The CSE program is well documented and can be found online at:

- For current students: <http://www.cs.washington.edu/students/ugrad/>, which summarizes in detail the program requirements and advising process.
- For prospective students, http://www.cs.washington.edu/prospective_students/undergrad/, which serves both an educational role ("Why Choose CSE?") and provides information on how to prepare for and apply to the major.

The undergraduate advising team also manages the 5th year masters (Combined BS/MS) program. There is a separate Graduate Advising Office supervised by another faculty coordinator for our other two graduate programs.

The CSE Advising Office supervises both the computer engineering and computer science majors. All advisors are knowledgeable about both degree programs and can advise any CSE student. The program information for the two majors is closely coordinated to take advantage of the commonality of the two degrees.

D.2 Career Guidance

Since our last ABET review, we have developed a concrete career preparation component for our students. Some of our events include:

Employer Panel (Fall Quarter)

The Employer Panel is the first CSE recruiting preparation event of the year. The panel consists of four employer representatives: one recent grad/engineer, one large company HR rep, one small company HR rep, and one hiring manager. The purpose of the panel is to inform students about the recruiting process from the presenters' perspectives. To accomplish this, recent grads discuss their own job search, hiring managers discuss what they look for in candidates, and HR reps discuss how recruiting works within their companies. While the represented companies certainly provide some level of self-promotion, their focus is on providing a peek inside the recruiting process so the students will know how to best

prepare. This year's panelists were from Sony Media, Eggsprout, Microsoft, and Tableau Software.

Internship Panel (Winter Quarter)

The internship panel consists of recent CSE grads from companies such as Amazon, Zillow, Google and Microsoft who provide the insight CSE undergraduates need to secure the best-fitting internships, prepare for those internships, and know what to expect once they get there. Additional focus is on guiding students to make the best early decisions to pave the way for a successful transition into full-time employment in the future.

Resume Review Workshop (Fall and Winter Quarter)

In this workshop, HR reps and recruiters (or other people within companies who screen résumés or serve as the first or second-line reviewers) sit with small groups of CSE students for 15-20 minutes to critique resumes, offer suggestions, and help them refine the way they present themselves on paper. It is a chance for students to get honest feedback about what is working, what is not, and to see examples from other students to help them craft a polished and effective résumé

Mock Technical Interviews (Fall and Winter Quarter)

Mock technical interviews afford students the opportunity to participate in a one-on-one simulated interview. Engineers, hiring managers, and other technical employees from local software companies conduct the technical interviews which consist of whiteboard questions, problem solving puzzles and coding questions. Interviews last 30 minutes with an additional 10 minutes allotted for interviewers to give the students open feedback on how they did and what they can do to improve their technical interview performance.

Technical Interview Coaching (Fall Quarter)

This event is designed for CSE undergrads who will be interviewing for a full-time or internship position in the coming year. It is a preview of the types of technical interview questions they will likely encounter. Technical interviewers from the largest local high-tech firms meet with groups of 2-4 students. The interviewers describe their technical interview process, give students a few sample questions, and coach students on what they look for in answers. The sessions include sample programming questions, logic questions and puzzles. The experts also provide examples of what they ask in actual technical interviews and how to successfully answer to land the job. Students attend three separate 15 minute sessions with three different company volunteers.

E. Work in Lieu of Courses

With the exception of a limited number of co-op credits (ENGR 321), the University of Washington does not award course credit for work. A student with significant outside experience may be allowed to test out of a course by interviewing with the faculty lead in charge of that course. The student would then use a higher level course to satisfy the credits.

We take active steps to encourage our students to participate in co-op programs and other industry internships. Currently, roughly two-thirds of our undergraduates participate in co-ops or

internships. These students may be awarded up to 1 CSE Senior Elective credit per quarter, for up to 2 quarters, for participating in a co-op and completing additional work, including a presentation on their work experience. In 2011-12, 40 students pursued a co-op experience, working for 63 different companies ranging from startups to Google, Amazon, Facebook, Apple, and Microsoft. 70% of them engaged in a single (3 month) internship, 25% did the equivalent of two quarters of internship, and 5% completed three quarters. Assessment of these activities is addressed in Section 4A.

F. Graduation Requirements

The Computer Engineering program requires 180 credits to graduate. Students fulfill a Written & Oral Communication component, General Education component, a Mathematics and Science component, and a Computer Engineering component that consists of required core classes and senior electives. The specific courses required are detailed under Criterion 5, Curriculum.

Students may graduate at the end of any academic quarter. Satisfactory completion of the Bachelor of Science in Computer Engineering curriculum is assured by an auditing process and by encouragement to regularly meet with the advising staff. A special checkpoint meeting takes place two or three quarters prior to the proposed quarter of graduation. Each student must file a formal degree application with an advisor (who also checks with the University of Washington's Degree Audit Reporting System to ensure compliance). This meeting includes a plan of courses in progress and those scheduled for the remaining quarters leading up to graduation. The outcome is that each student is required to have an acceptable terminal course plan over two quarters in advance of their actual graduation.

CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES

A. Mission Statement

Our CE Mission Statement is:

To educate our students so that they will reach their full potential in computer engineering research and industrial practice through a deep understanding of the fundamentals of the field, their application in solving problems and creating products, and with an affinity for lifelong educational renewal.

B. Program Educational Objectives

Our mission statement leads to the following program educational objectives, which are available online: [www.cs.washington.edu /ABET/](http://www.cs.washington.edu/ABET/)

Engineering Quality: Our graduates will engage in the productive practice of computer engineering to identify and solve significant problems across a broad range of application areas.

Leadership: Our graduates will engage in successful careers in industry, academia, and public service and attain leadership positions where they have impact on their business, profession and community.

Economic Impact: Our graduates will enhance the economic well-being of Washington State through a combination of technical expertise, leadership and entrepreneurship.

Lifelong Learning: Our graduates will adapt to new technologies, tools and methodologies to remain at the leading edge of computer engineering practice with the ability to respond to the challenges of a changing environment.

C. Consistency of the Program Educational Objectives with the Mission of the Institution

The departmental program objectives are completely consistent with those of the College of Engineering and the University. The key sentence of the University of Washington's mission statement (<http://www.washington.edu/admin/rules/policies/BRG/RP5.html>) is "The primary mission of the University of Washington is the preservation, advancement, and dissemination of knowledge." These correspond to our engineering quality, leadership and economic impact, and lifelong learning objectives. The key part of the College of Engineering's statement (<http://www.engr.washington.edu/about/mission.html>) is "working to dramatically improve the

quality of life in our state, our nation, and our world.” Our program objectives address that goal in both its economic and education and knowledge interpretations.

D. Program Constituencies

Our primary responsibility as part of the University is education and our most important constituents are:

- students currently enrolled in our major
- our alumni

The organizations that directly rely on the students we graduate and their knowledge, abilities and skills form another set of constituents:

- prospective employers, most importantly the member companies of our Industry Affiliates Program. Our Affiliates Program currently comprises more than 100 companies, from industry leaders (Amazon.com, Google, Intel, Microsoft) to startups, plus several national labs and several venture capital firms. The current list is online at http://www.cs.washington.edu/industrial_affiliates/current/. There is an annual Affiliates meeting where representatives from member companies interact with faculty and students to learn about new research results, forge new research relationships, interview prospective students for internships and permanent positions, and give advice on both research and educational issues.
- the nation’s leading university graduate programs.

E. Process for Revision of the Program Educational Objectives

Each of our constituencies participates in the process of reviewing our educational objectives and evaluating our success in meeting them. Our process for this is to engage all our constituencies in feedback loops that take advantage of their vantage points. While we directly present our PEOs for review and comment to our industrial affiliates, for the most part our process relies on our department leadership to distill all the inputs we receive into establishing and revising those objectives. Updates are primarily driven by the feedback we have received, although we also do periodic reviews. The most recent periodic review was performed by the department Executive Committee during Spring 2013, and did not result in any changes to our PEOs.

We obtain relevant feedback by means of surveys, interviews, and review sessions. Interaction between constituencies, for example between students and employers, is also part of the process. Through both our Co-operative Learning and Affiliates programs, we receive direct feedback on how students are meeting our current program objectives, and about any deficiencies that might suggest changes to those objectives. The fact that we are one of the top suppliers in the nation to Amazon.com, Google, and Microsoft, as well as being the predominate supplier to small and mid-size companies in the Puget Sound region, means that CSE leadership is in close touch with recruiting managers and engineers at these companies, providing a tight feedback loop.

Alumni working at regional tech companies, large and small, are closely engaged with current students. We invite them back to conduct resume review workshops, mock interviews, and a popular one-quarter seminar focused on “success factors” in the tech workforce. Additionally, the vast majority of our students participate in one or more summer internships. All of these are mechanisms for additional tight feedback on our current PEOs and possible deficiencies.

Our current students meet with the department chair each quarter. During these meetings students ask questions and raise concerns about the program. Examples of problems that have been raised and addressed this way include infrastructure issues in the labs, the workload in some courses, the need for classes that cover increasingly important topics like security and HCI, and the choice of tools used in our courses and their relationship to industrial practices. These meetings also allow the chair to inform students of new initiatives and changes and solicit questions and comments. This ensures that students are current on departmental plans and that we know about issues that concern the students.

Finally, the Undergraduate Advising Office and the Undergraduate Program Coordinator are continually in contact with our primary constituencies and forward issues that merit attention to the Curriculum Committee. As well as direct contact with students, advising provides a department-level anonymous feedback channel that any of the constituencies may use. When problems are identified, the Undergraduate Program Coordinator and the advisors work with the relevant faculty to produce a workable solution and, if needed, bring it before the Curriculum Committee and possibly the entire faculty.

CRITERION 3. STUDENT OUTCOMES

A. Student Outcomes

Our mission statement, program objectives, and student outcomes are all publicly available at <http://www.cs.washington.edu/ABET/>.

Our student outcomes comprise the set of skills and abilities that our curriculum is intended to give our students so they can meet the educational objectives we have set for them:

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve computer engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice.

B. Relationship of Student Outcomes to Program Educational Objectives

We have four broad program educational objectives. We examine each in turn and indicate which student outcomes are the most directly relevant to achieving the program objectives.

Engineering Quality: *Our graduates will engage in the productive practice of computer engineering to identify and solve significant problems across a broad range of application areas.*

All of our student outcomes contribute to this objective, so we do not enumerate them here.

Leadership: *Our graduates will engage in successful careers in industry, academia, and public service, providing technical leadership for their business, profession and community.*

Leadership can take a number of forms, as indicated by the statement of our objective. Particularly relevant to this objective are these outcomes:

- (c) an ability to design a computing system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve computer engineering problems
- (f) an understanding of professional and ethical responsibility
- (h) the broad education necessary to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice.

Economic Impact: *Our graduates will enhance the economic well-being of Washington State through a combination of technical expertise, leadership, and entrepreneurship.*

This objective is closely related to the previous one, but focuses specifically on economic impact setting a lower bar on leadership activities. Most particularly relevant outcomes are:

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (d) an ability to function on multi-disciplinary teams
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice.

Lifelong Learning: *Our graduates will adapt to new technologies, tools and methodologies to remain at the leading edge of computer engineering practice, with an ability to respond to the challenges of a changing environment.*

Particularly relevant are:

- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice.

Our alumni survey results have consistently shown very good achievement on this objective.

CRITERION 4. CONTINUOUS IMPROVEMENT

A. Student Outcomes

A.1 Data Sources

We employ a wide range of processes for collecting data on student outcome attainment. Our processes span a spectrum from narrowly focused on individual outcomes to more open ended, and involve all of our constituencies. Table 4.1 lists these data sources. Table 4.2 identifies the primary data sources used to assess each of our student outcomes, as well as the most important secondary sources of information.

Table 4-1 Information sources for assessment

<i>Assessment Process</i>	<i>Description</i>	<i>Frequency</i>	<i>Where Maintained</i>
Targeted Course Work	Instructors establish, assess, and evaluate performance indicators.	Regularly	Online on CSE servers
Affiliates Feedback	Special feedback session held at our yearly Affiliates Meeting	Yearly	CSE Advising
Co-op student and employer feedback	Students perform pre- and post- co-op self-evaluations. Employers independently rate using the same criteria.	Each co-op	COE Co-op & Internship Program and CSE Advising
Alumni Survey	A web-based survey administered by CSE	Yearly	CSE Advising
End-of-program Survey	Either group interviews or a web-based survey of about-to-graduate students. Administered by the UW Center for Instructional Development (CIDR).	Yearly	CIDR and CSE Advising
Exit Survey	Administered by CSE.	Yearly	CSE Advising
Regular Course Evaluations	All courses are evaluated. Evaluations include standard question set by the University, outcome-related questions set by CSE, optional questions set by the instructor, and free-form comments.	Regularly	UW Office of Educational Assessment and CSE Chair's Office
Targeted Course Work	Individual assignments and exams, or portions of assignments or exams, may be targeted to assess particular outcomes.	Regularly	Instructors. We're migrating onto a CSE-hosted web service.
Faculty Self-Evaluations	Instructors write reflective statements on each course they teach.	Every course offering	Online, by CSE.
Capstones	Capstones are showcases for many student outcomes.	Regularly	CSE Instructors; Dept. Videos

<i>Assessment Process</i>	<i>Description</i>	<i>Frequency</i>	<i>Where Maintained</i>
Per-Course Instructor Meetings	As part of our curriculum revision, we have held per-course post mortems where instructors of each new course convene and talk about what is working and what isn't.	Yearly	Outcomes are incorporated into course syllabi, which are maintained online.
Full-Faculty Curriculum Meetings	The full faculty has convenes to consider cross-course and full-curriculum issues.	Yearly	Outcomes are incorporated into course syllabi, pre-reqs, and graduation requirements.

Table 4-2 Assessment procedures for student outcomes

<i>Student Outcome</i>	<i>Primary Data Sources</i>	<i>Significant Additional Sources</i>
(a) an ability to apply knowledge of mathematics, science, and engineering	Targeted Course Work	Alumni survey Employer feedback
(b) an ability to design and conduct experiments, as well as to analyze and interpret data	Targeted Course Work Capstones	Employer feedback
(c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability	Targeted Course Work Capstones	Alumni survey
(d) an ability to function on multi-disciplinary teams	Targeted Course Work Capstones	Employer feedback End-of-program feedback
(e) an ability to identify, formulate, and solve engineering problems	Targeted Course Work	Alumni survey
(f) an understanding of professional and ethical responsibility	Targeted Course Work	Alumni survey
(g) an ability to communicate effectively	Targeted Course Work Capstones	Employer feedback Alumni survey
(h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context	Targeted Course Work	Alumni survey
(i) a recognition of the need for, and an ability to engage in life-long learning	Targeted Course Work	Employer feedback Alumni survey
(j) knowledge of contemporary issues	Targeted Course Work	Employer feedback
(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	Targeted Course Work	Employer feedback Alumni survey

A.2 Assessment Process Overview

Figure 4-1 gives a high level overview of our assessment and evaluation procedures. At this level of detail, the figure describes both our processes before the curriculum update and our processes during it, with one exception: we involved our External Advisory Committee (EAC) in our recent curriculum revision, whereas they would not normally spend time on smaller curricular matters.

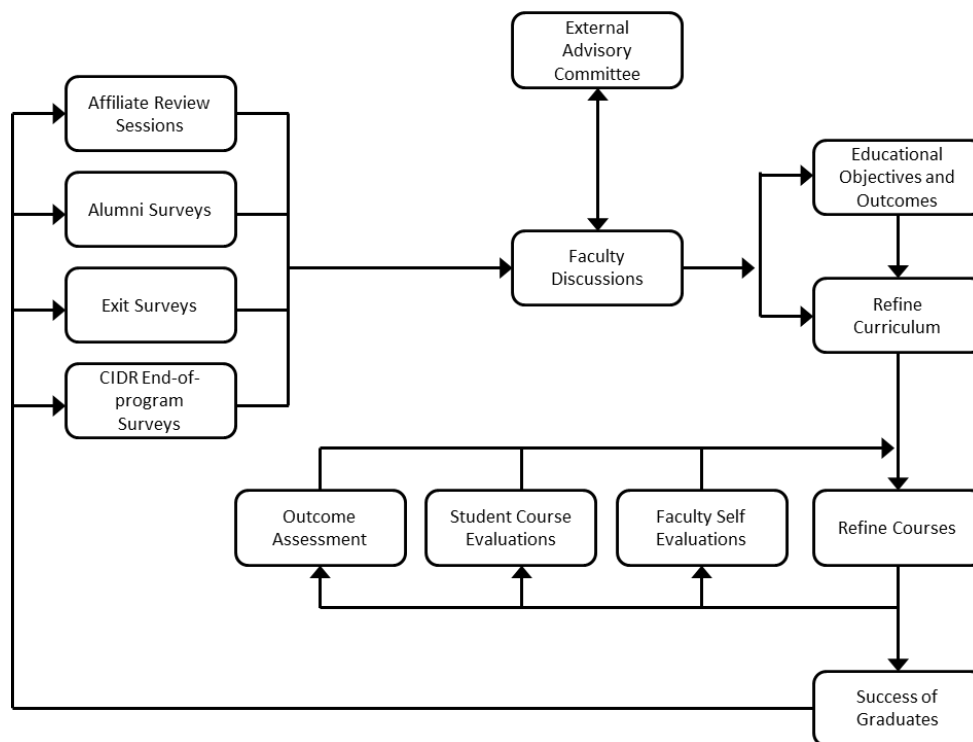


Figure 4-1 Summary of Department Assessment and Improvement Processes

There are two primary feedback loops in our process, one best suited to quickly refining individual courses, and another suited to making improvements that cut across courses. The large feedback loop evaluates the program as a whole from the standpoint of our objectives and how well our curriculum prepares students to achieve these objectives. The loop uses feedback from graduating students, alumni and our industrial affiliates. We ask an outside group, the Center for Instructional Development and Research (CIDR), to interview our soon-graduating students. Additionally, we run an online exit survey that includes both qualitative and quantitative assessments of our program by our graduating students, as well as an annual alumni survey soliciting feedback from our students after they have experience in the workplace or graduate school. Finally, we get feedback from our Affiliates on a regular basis on how well we have prepared our students and where we can make improvements.

This feedback is used to evaluate the effectiveness of the overall curriculum in meeting our objectives and student outcomes. The Undergraduate Program Coordinator, together with the Undergraduate Advising Office and the Curriculum Committee, continually evaluates the overall program. There is no set timetable for evaluation – it is ongoing. However, the staff of the Advising Office meets weekly with the Undergraduate Program Coordinator to discuss issues

that have arisen. The Curriculum Committee meets once or twice per quarter to discuss curriculum changes as well as issues forwarded by the Undergraduate Advising Office that are likely to require further attention and further faculty discussion. During the transition to the new curriculum, we also held meetings of faculty who have taught each individual core courses, to share experiences, discuss possible problems and solutions, and to refine and document course content, outcomes, and expectations. It is part of our standard procedure to involve faculty in this process through discussions at our annual faculty retreat and regular faculty meetings. This process has led to many changes in our curriculum and the way we run the program.

The smaller feedback loop in the figure above is used to evaluate the effectiveness of individual courses in contributing to the overall curriculum objectives and achieving student outcomes. Faculty members perform most of the assessment, of the course itself, of their own and each other's performance in leading courses, and of student outcomes. Officially, assessments are coordinated by the advising staff and Undergraduate Program Coordinator, which may escalate issues if they are not being satisfactorily dealt with, but most often faculty teaching a particular course will interact to discuss their experiences.

We evaluate student outcomes using performance indicators, which are established on a per-course basis. Faculty devise targeted assessment procedures to assess the performance indicator, and use them to quantify student achievement using four categories: Exemplary, Satisfactory, Improving, and Novice. Outcomes are recorded online, and we are developing web pages that allow convenient tracking of outcome achievement over time.

In addition to this direct outcome assessment procedure, we also have several indirect procedures. Currently enrolled students give us feedback in many ways and at several points in their student career. Part of this feedback is explicitly organized. For example, all students evaluate our teaching and courses every quarter, and this feedback is used by instructors to improve the course material and the way it is delivered, and by the department to identify courses that may require special attention. Instructors also write self-evaluations of their courses which are included in annual peer reviews of teaching. We have found this feedback invaluable for improving teaching and sharing ideas among faculty.

Other feedback is more informal. Students interact with faculty and advising staff through individual meetings and many special events, such as our quarterly undergraduate lunches and other sessions on specific topics such as research/graduate-school/career planning organized by the advising staff. The lunch meetings with the Chair and Undergraduate Program Coordinator provide the opportunity for students to raise educational concerns and for the department to respond in a timely manner. Students may bring complaints and concerns to the undergraduate advising staff, who collate them and bring them to the attention of the department. Most course Web sites, as well as the advising Web site, provide an anonymous feedback form for student use. Comments, complaints and suggestions and sometimes compliments reach the Undergraduate Program Coordinator and, through that position, the Curriculum Committee, Executive Committee, and/or Chair as appropriate. Depending on the seriousness of the issue, it may be dealt with by the advising staff, the Undergraduate Program Coordinator, or passed along to the faculty as a whole for discussion and solution.

We also use input received from other groups. Information about prospective students reaches the faculty primarily through the undergraduate advising staff through quarterly information sessions and individual appointments with prospective majors. Regular meetings and workshops

with Community College instructors provide important feedback on our introductory curriculum (which is mirrored at the state's CCs). In addition, we interact with the University's admissions office when we admit freshmen directly to our major. Through the advising staff's information sessions and our undergraduate admissions process, the department can make adjustments to pre-requisites to the major at UW. We have extensive outreach activities at the K-12 level.

A much larger audience reviews the videos we produce to capture the resulting projects from many of our capstone courses. The audience includes the general public (the videos are broadcast on the UW cable television channel) and prospective students. Comments from all these groups make their way back to the Undergraduate Program Coordinator and the Advising Office and are incorporated with other feedback.

Finally, we are just this year getting involved in massive online education, offering five courses (Programming Languages, Computer Networks, The Hardware/Software Interface, Data Science, and Computational Neuroscience) via Coursera. We are eager to gain experience with education at this scale, and excited by the prospect that experience with and feedback from a huge number of students can be fed back into our program to improve the experiences of our local students. More information is available at

<http://www.cs.washington.edu/education/multimedia/moocs>.

A.3 Targeted Assessment of Outcomes in Courses

The most direct assessment we perform of student outcomes is through targeted assessment in courses. Instructors define performance indicators for the student outcomes addressed by their courses. They then use those indicators to categorize student achievement in four levels: Exemplary, Satisfactory, Improving, and Novice. The instructor also selects representative work samples from the top, middle and bottom of the class and saves these. This additional information allows assessments by different instructors to be evaluated and compared.

We perform targeted assessment on an ongoing basis, using a sampling approach in which we assess all outcomes each year, but using only some courses. Our goal is to assess each outcome at least twice per year, and to cycle among the courses involved in assessments on about a three year rotation. Table 4-3 shows the assessment schedule used during the most recent academic year. The coverage of an outcome reflects the relative attention we currently believe it requires.

Table 4-3 Academic Year 2012/13 Targeted Assessment Schedule

<i>Outcome Assessed</i>	<i>12au</i>	<i>13wi</i>	<i>13sp</i>
a) an ability to apply knowledge of mathematics, science, and engineering	CSE 461		CSE 352
b) an ability to design and conduct experiments, as well as to analyze and interpret data	CSE 461	CSE 454 CSE 460 CSE 481I	CSE 477 CSE 481D
c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability	CSE 461	CSE 454 CSE 460 CSE 481I	CSE 477 CSE 481D

Outcome Assessed	12au	13wi	13sp
d) an ability to function on multi-disciplinary teams		CSE 454 CSE 460 CSE 481i	CSE 477 CSE 481D
e) an ability to identify, formulate, and solve engineering problems	CSE 461		CSE 444
f) an understanding of professional and ethical responsibility		CSE 484	CSE 351
g) an ability to communicate effectively		CSE 454 CSE 460 CSE 481i	CSE 477 CSE 481D
h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context	CSE 466		CSE 403
i) a recognition of the need for, and an ability to engage in life-long learning	CSE 461		CSE 451
j) knowledge of contemporary issues		CSE 484	CSE 403
k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	CSE 461		CSE 461

We have developed, and are continuing to enhance, a set of web-based tools for collection and review of the assessment materials. This is a department activity. Assessment materials for all courses are available to all faculty, as well as the advising staff.

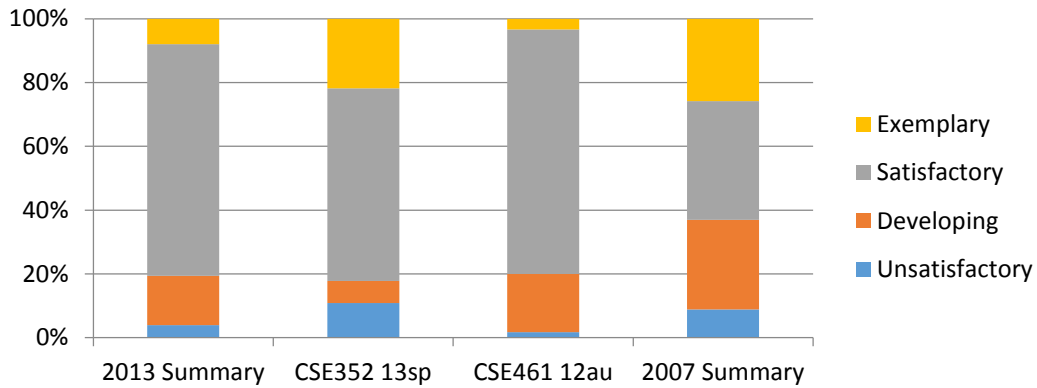
In evaluating the assessments, our standard is an average outcome of at least 80% in the Exemplary or Satisfactory categories. This standard reflects the desire for a high level of achievement while recognizing that most of the measurements will be taken perhaps substantially before students are ready to graduate.

Table 4-4 contains graphs showing the most recent targeted assessment results, one graph per student outcome. In each graph we show:

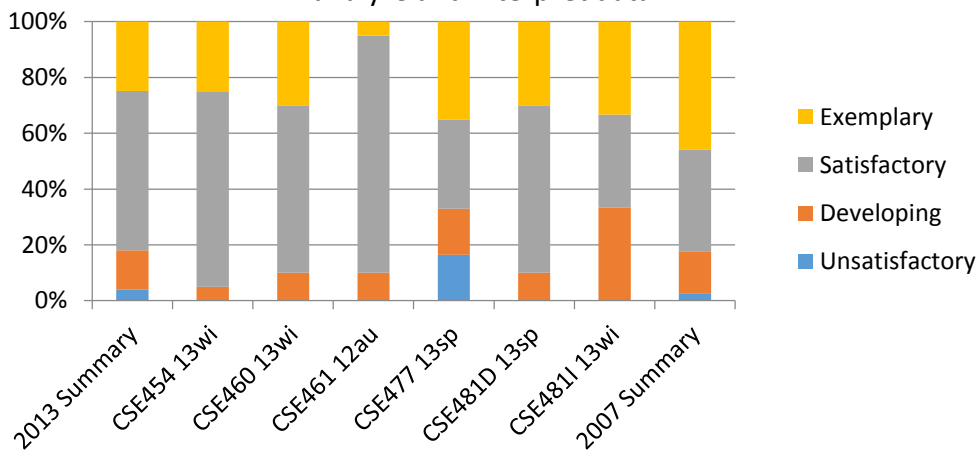
- The summary assessment results for that outcome from 2007
- Individual course assessment results from the 2012-13 academic year. Each course uses from one to three performance indicators for each outcome. The results shown for an individual course average the results overall performance indicators used by that course. By looking at the individual course results we get a sense of the amount of variation due to distinctions among instructors, as well as how advanced the course may be.
- The 2012-13 assessment results averaged over all courses assessing that outcome.

Table 4-4 Targeted Assessment Results

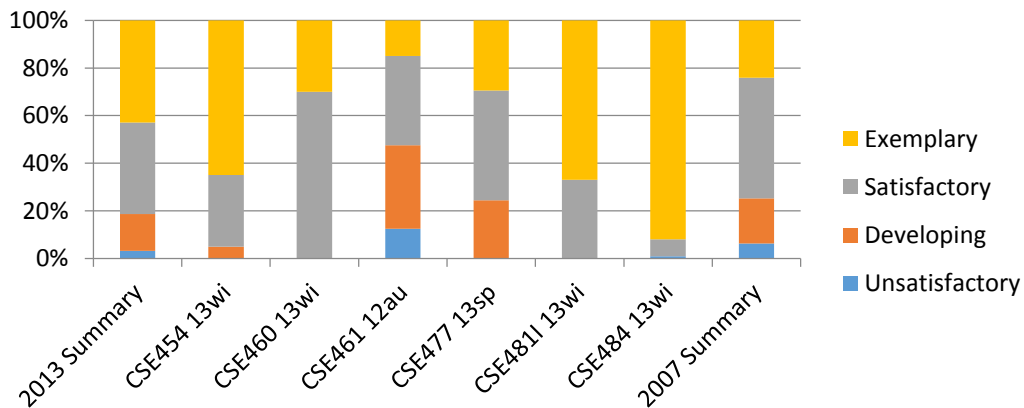
a) an ability to apply knowledge of mathematics, science, and engineering



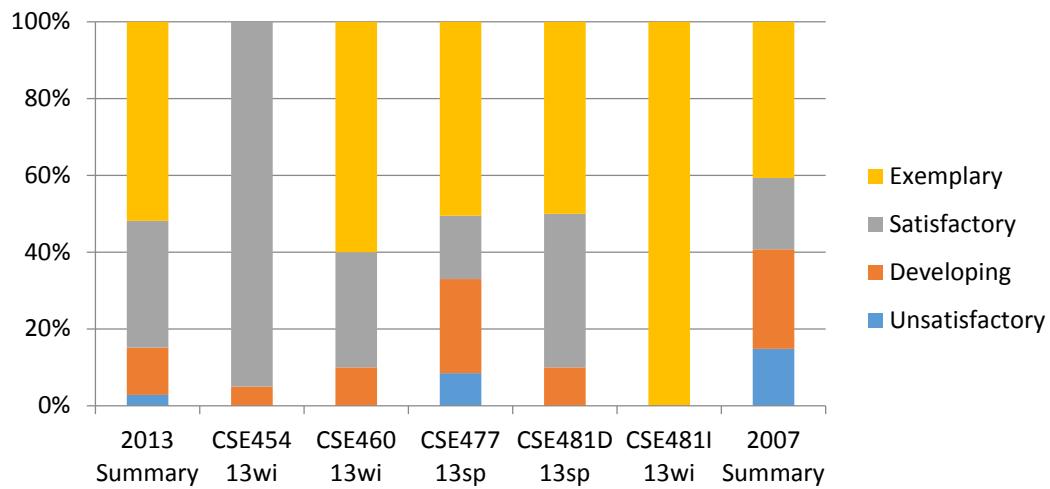
b) an ability to design and conduct experiments, as well as to analyze and interpret data



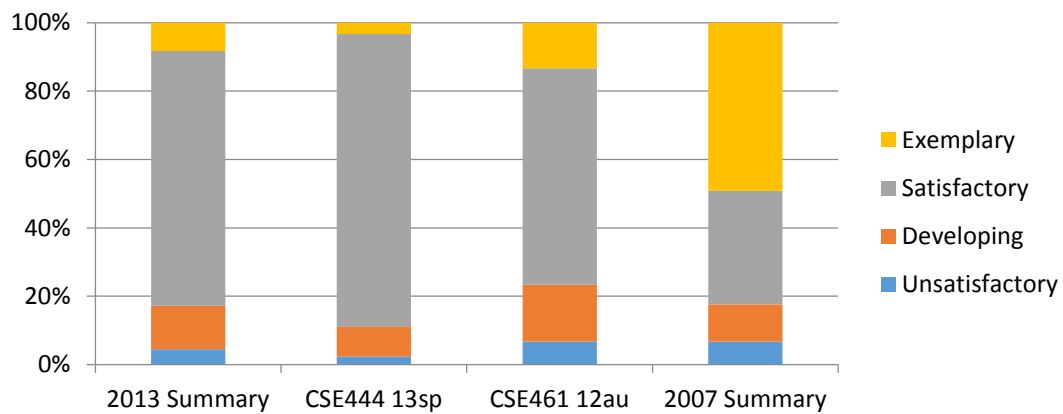
c) An ability to design a system, component, or process to meet desired needs within realistic constraints



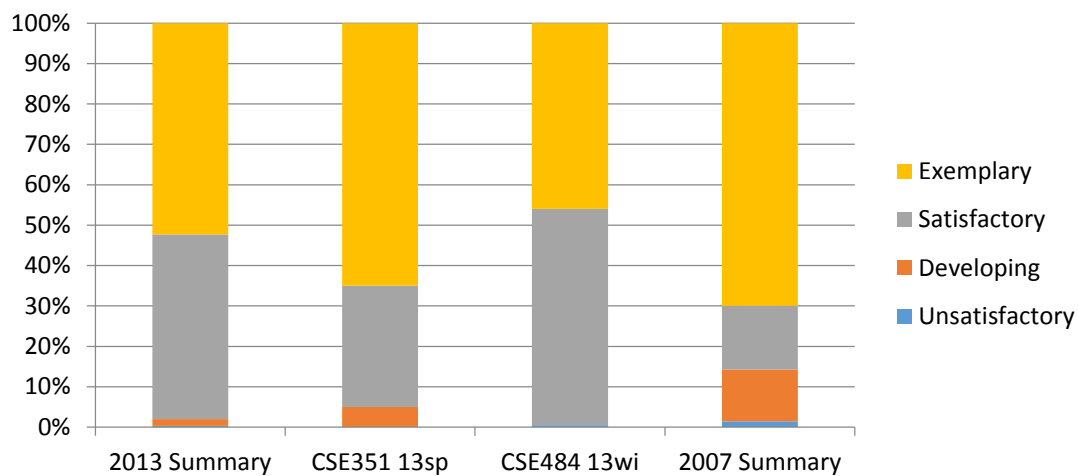
d) an ability to function on multi-disciplinary teams



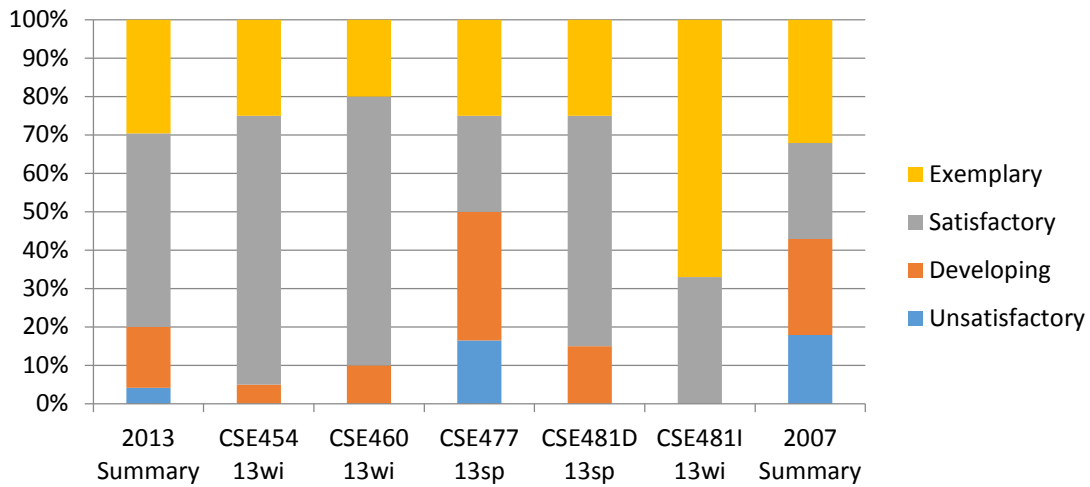
e) an ability to identify, formulate, and solve engineering problems



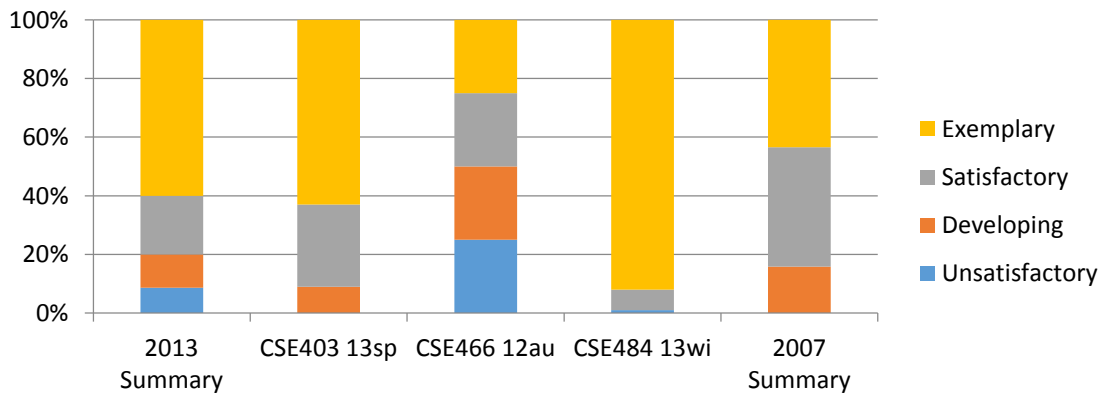
f) an understanding of professional and ethical responsibility



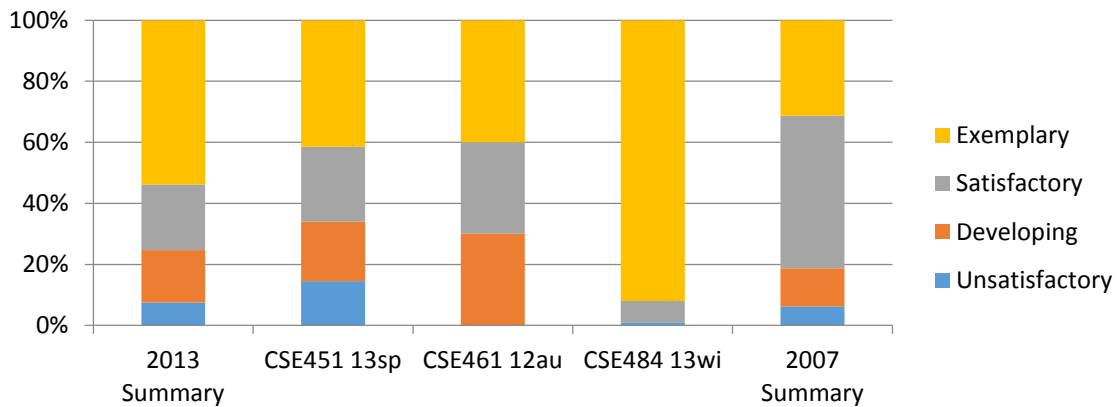
g) an ability to communicate effectively

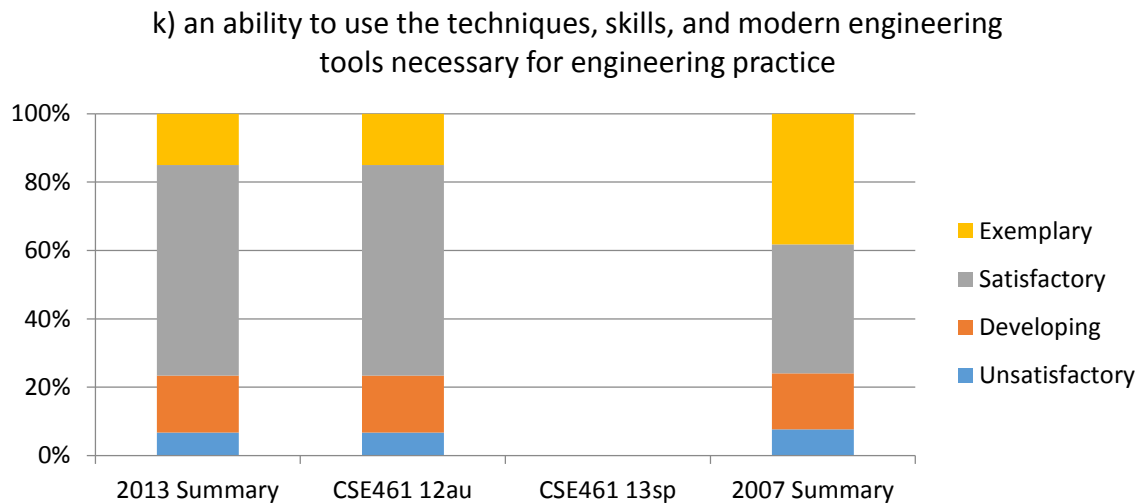
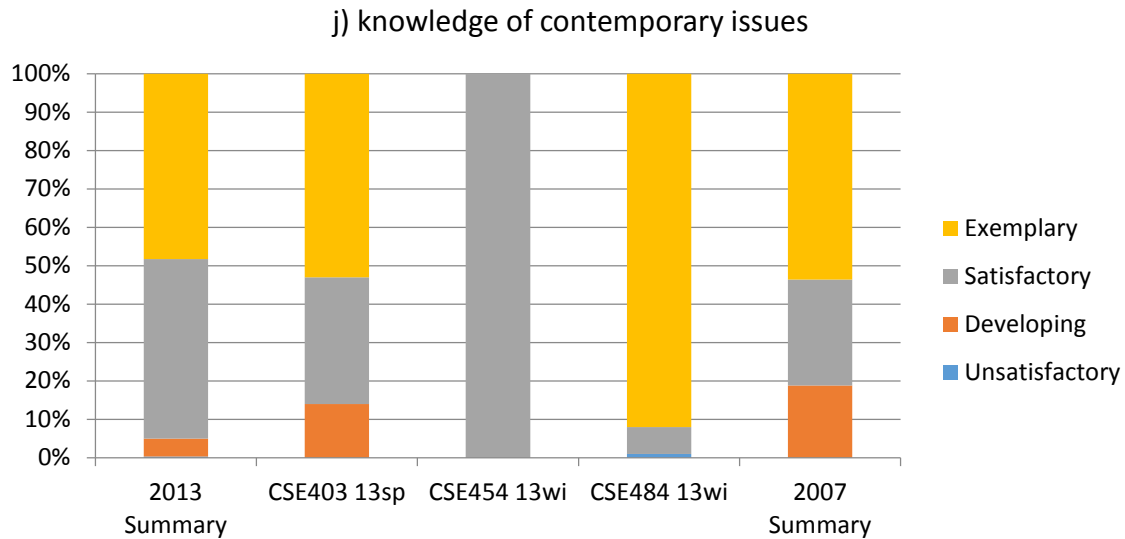


h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context



i) a recognition of the need for, and an ability to engage in life-long learning





Looking over these results we see that for the most part we are meeting our goal that 80% of our students achieve at the Satisfactory level or higher. The exceptions ((i) and (k)) are both very near the goal, and neither is particularly alarming. For (i), the wide variation in the individual course assessments indicates that there is likely a reasonable amount of uncertainty in the sample average. For (k), at the time we're submitting this report we have results from only a single class. Nonetheless, we'll do a more in depth analysis of these results in the Section B.

A.4 Additional Assessment Processes

A.4.1 Affiliates feedback

Current and prospective employers of our students are strongly encouraged to join our department's Industrial Affiliates Program and attend an annual meeting. We have strong ties to these companies (including Microsoft, Intel, Google, Amazon, IBM and others - http://www.cs.washington.edu/industrial_affiliates/current/). This key set of stakeholders of

major and local companies hire the bulk of our students. We take advantage of our Industrial Affiliates Program to obtain feedback on our students.

We hold an annual two day meeting for our industrial affiliates. Part of this event is a session focused on curricular issues. Affiliates have the best vantage for observing how well our students are meeting our educational objectives. It is often easier for them to assess the more concrete issues of student outcomes (and they tend to be more interested in those because they are directly pertinent to their immediate business interests), and so typically the discussion relates most directly to student outcomes.

Our most recent meeting was held in October 2012. Although we are very interested in getting feedback on the downstream effects of our new curriculum (and indeed asked the affiliates about what they were seeing), it was still too soon for employers to have enough experience with students who have gone through it; we should start seeing that next year.

We have two general questions we pose at these meetings: what are the strengths and weaknesses you have seen in our students, both in interviews and as employers; and are you seeing things in students from other institutions that might suggest we should look at their curricula for potential improvements to ours? For the latter, there was mention of the Northeastern and Waterloo co-op programs, and the fact that they afford longer internships.

About weaknesses in our students, the main feedback this year had to do with the level of C programming experience they have. This is something addressed by the revised curriculum: it includes a new course, CSE 333, which concentrates on C/C++ and its use in systems programming, the kind of applications most often mentioned by our affiliates. We hope to see this improvement reflected in affiliate's feedback as early as next year.

A.4.2 Co-op feedback

Approximately 80% of our students participate in at least one internship, and many through the Co-Op program itself. Feedback takes two forms. First, students fill out a survey in which they rate their achievement level in 17 detailed areas, shown in Table 4-5. These questions are essentially the student outcomes, although in some cases a single outcome has been further broken down into more specific components. Students fill out the same survey both before beginning and after completing the co-op. In addition, their employer fills out the survey. Finally, the employer is invited to provide free form comments.

Table 4-5 Co-op Survey Categories

- | |
|---|
| <ol style="list-style-type: none">1. Ability to apply engineering theory and principles to the work environment.2. Ability to design a system, component, or process to meet desired needs.3. An understanding of professional and ethical responsibility.4. Ability to communicate effectively on an interpersonal level.5. Ability to communicate effectively in formal presentations.6. Ability to communicate effectively in technical writing.7. Recognize the need for lifelong learning.8. Ability to use techniques, skills and modern engineering tools necessary for engineering practice.9. Ability to design and conduct experiments.10. Ability to analyze and interpret data.11. Ability to function on multi-disciplinary teams. |
|---|

12. Knowledge of contemporary issues in your discipline.
13. Ability to identify, formulate and solve engineering problems.
14. Recognize the value of hands-on experience.
15. Ability to think critically.
16. Ability to manage change.
17. Understanding of office dynamics.

Because co-ops occur during a student's work towards a degree, we expect co-op feedback to reflect the real-world impact of our curriculum changes earlier than feedback taken further downstream, such as our alumni survey data or affiliates feedback.

Figure 4-2 shows average survey results for co-ops that took place between Spring 2011 and Summer 2012. There are a number of ways to interpret this data. First, our students largely underestimate their abilities, compared to the impression their employers have, except in areas related to communication and running and evaluating experiments. Second, those areas are where students and employers agree that our students are weakest. Furthermore, these are areas students have identified for possible improvement in other feedback, for instance, in quarterly lunches with the department chair and in surveys of graduates. (Planning how to address this is a high priority item for the coming year.) Finally, the students perceive only incremental improvement in their skills due to the co-op experience. This is at odds with direct faculty observation, which indicates that the co-op experience brings a level of maturity to the students' work that is much more difficult to achieve without doing a co-op.

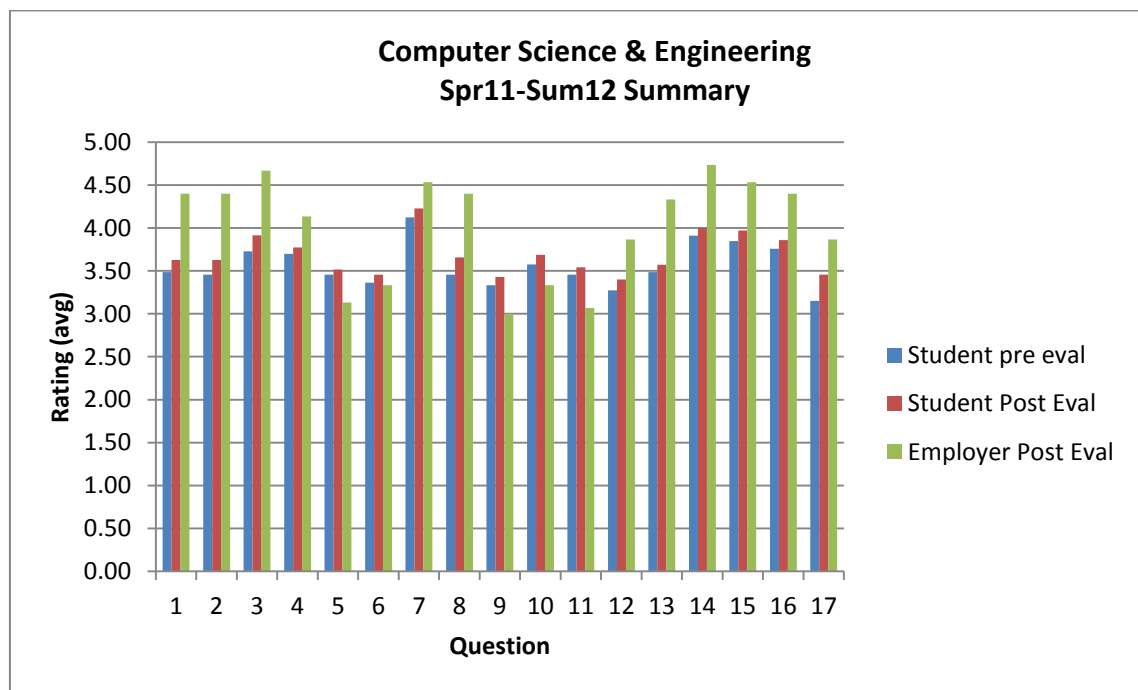


Figure 4-2 Co-op experience survey 2011/2012

In addition to the quantitative data just shown, we also make use of the free form comments by employers. We cite here some typical examples from 2012, selected because they reflect directly on our four program educational objectives, as listed in Section 2.B:

Engineering Quality

“(student) did a great job for us. He was very focused on delivering what was asked of him. His growth over the last 6 months has been fantastic.”

Leadership

“(student) did a great job taking on complex design tasks and we have employed some of his engineering into our customers production environment.”

Economic Impact

“(student) is a pleasure to work with and we look forward to having him back on our team once he completes his degree. Both his work ethic and his ability to quickly learn new concepts made (student) an important part of the team and a valued employee.”

Lifelong Learning

“(student) came to us with a solid knowledge base and has learned even more on the job. If he runs into a technical issue that he doesn’t immediately know how to solve, he has been able to find the resources quickly, learn new tools, and apply them to our project.”

A.4.3 Alumni surveys

We survey our alumni annually. This provides input from the vantage point of several years in industry or graduate school, and so is qualitatively different from most of our other assessments. Alumni are asked questions that directly reflect upon our educational objectives and the student outcomes that support them.

The most recent survey was performed in November 2012. 202 former students responded to this survey. Only 2.5% of them had followed the new curriculum entirely, and 4.5% of them a mixture of the old and new. The following tables summarize the responses:

Years since graduation	Number
0-5	96
6-10	46
>10	60

Work Location	Number
Washington State	101
US	65
Foreign Country	5

Current Position	Number
Masters Student	2
PhD Student	12
Other Student	2
Developer/Engineer	121
Engineering Management	17
Executive	20
Currently Unemployed	1
Other/No Response	27

Company	Number
Microsoft	12
Google	11
Amazon.com	21
Adobe	5
Other	122

Professional Development	Number	Advanced Education	Number
Attended conferences	114	Currently a student	16
Authored papers	51	M. S. Degree	37
Learned new programming language	143	Ph.D. Degree	23
Other, e.g. in house training	18	Other Degree	5
Member of professional society	47		
Certificate Programs	18		
Online/Non-Degree Courses	47		

The graduate schools attended include MIT, Carnegie-Mellon, Stanford, UCSD, University of Maryland, and University of North Carolina. In addition, 58 students indicated they planned to return to school for a Master's degree in the future.

On the whole the comments about the department and their experience here were very positive. When asked about what, in retrospect, the department did particularly well students mentioned excellent teaching and facilities, a good balance of theory and practice, a "hardcore" engineering education with many lab experiences, the coordinated curriculum, access to research experience, being taught how to learn, strong capstone courses, and exposure to a broad range of topics.

When asked what the department should improve, students commented that it would have been useful to have larger group projects, more emphasis on internships and research experiences, more practice on presentation skills, more advanced courses, and larger and more complex software engineering projects. These comments are largely consistent with feedback we receive from employers after co-op experiences with our students, as well as from our current students. Some of the issues raised have been addressed since the students graduated, through introduction of new courses and changes to the kinds of work commonly assigned in our courses. Others are issues that we are sensitive to and are trying to address within the constraints of the undergraduate experience.

A.4.4 End-of-program interviews

Starting in 2005, we have asked staff from the UW Center for Instructional Development and Research (CIDR) to interview our soon-to-be-graduating students. Current students are surveyed by staff from the Center for Instruction Research, either through live group interviews or by web-based surveys, near the end of the program. Based on their experience with the two, CIDR reports different advantages to each, but feels that the online surveys may give a more accurate picture of student sentiment since it eliminates social pressure that might arise to agree with the majority in a group meeting.

The questions CIDR asks focus on the extent to which the students feel they have successfully met the student outcomes we set. Students are asked how well they think the department has helped them achieve each of our outcomes (a)-(k) (Section 3.A). Students rank the contribution on a scale of 0 to 5. The student responses for the 2 years preceding our curriculum update and the most recent two years are shown below in Figure 4-3. For the 2012 results, about 34% of the 170 respondents graduated under the old requirements (they likely took no new courses), 37% under the transitional requirements (they likely took a mixture of old and new courses), and 29% under the new requirements.

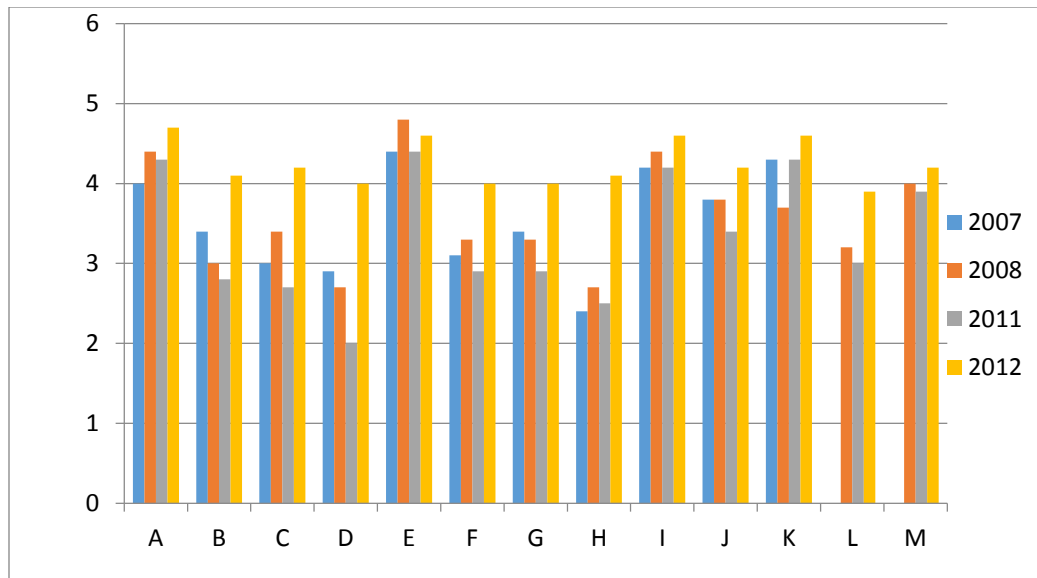


Figure 4-3 Student assessments of outcomes from end-of-program interviews

While the sample size here is too small to be convincing, the results are encouraging. The 2012 results dominate earlier ones in all cases (except for one small exception in category (e), where the distinctions are small across all years). In many cases the improvement is very large, much larger than the variation across the other three years' results, suggesting that this is not just random fluctuation. While both the 2011 and 2012 results are from after the curriculum revision, students graduating in 2011 have taken courses mostly from the old curriculum, while the 2012 graduating class have a mixture of both. Moreover, students graduating in 2012 are much more likely to have experienced 400-level courses that had started to react to the changes in the 300-level core curriculum. Thus, we are optimistic that our recent curriculum revision is having a strongly positive effect.

This end-of-program interview process also solicits specific comments and suggestions for improvement from the students. The responses are then collated and anonymized by CIDR and reported to the Undergraduate Program Coordinator, and from there go the department chair, executive committee and curriculum committee. We have found this process valuable for uncovering issues that we have been able to address quickly before they became any larger. A few recent examples of direct action taken in response to these sessions were:

- 2006-2008: Our students had been asking for a combined BS/MS degree for many years, so in 2008 we implemented a new program to meet this request.
- 2010-2011: Students wanted more experience with larger-scale software projects. We introduced CSE 331 (Software Design and Implementation), which is helping meet that need early in the curriculum.
- 2010 -2011: Students pointed out that an unintended consequence of the computer engineering degree's tracks (hardware vs. software orientations) was that they couldn't take a set of courses balanced between the two. We saw this as part of the larger issue of allowing a greater degree of flexibility, in light of the exploding diversity in applications of computer engineering. In Autumn 2011 we introduced a more flexible

Computer Engineering degree structure that allows students to mix and match between software and hardware subjects.

In 2012, the single improvement most often suggested by graduating students is to make a larger variety of courses available. This aligns well with one of the intents of our curriculum revision, to provide greater flexibility in course selection to students. It also aligns nicely with recent dynamics in the department: we hired six new faculty in 2012, bringing both additional teaching capacity and a significant jump in the variety of courses our faculty is prepared to teach.

A.4.5 Student surveys upon exit from the department

Graduating students are asked by CSE to complete a Web-based survey. One question we ask is where they will be employed or attending graduate school, as a way of ascertaining the downstream destinations of our students. From the 2011-2012 class, we were able to track 96% of our graduates: 20 are working at Microsoft, 21 at Amazon.com, 6 at Facebook, and 11 at Google. 18 are in graduate school, including MIT, CMU, Brown, Iowa State, Harvard, and our own department. We tracked 90% of our graduates from the 2010-2011 class: 30 took jobs at Microsoft, 5 at Google, 15 at Amazon, 90 at other companies such as Zillow.com, Amazon, and Tableau, and 26 went on to graduate school.

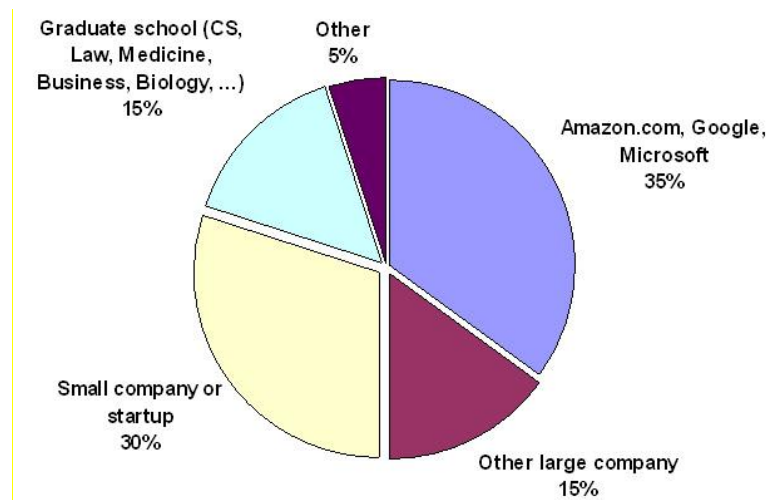


Figure 4-4 Graduate Exit Survey Destination Data for Undergraduates (2010-2012)

This exit survey is meant to supplement the one administered by CIDR: the CIDR survey focuses on student outcomes, while the CSE survey asks for broader feedback on students' general satisfaction with the experiences and courses provided in CSE. Student comments are collated by the lead advisor and reported to the Undergraduate Program Coordinator and the department chair. We use the departmental exit survey as one of many tools to proactively detect problems with our program.

Comments on the department are almost uniformly positive. Students are happiest with their senior level courses and capstones, and less satisfied with courses taken outside the department. We have made changes based on this input. Past exit surveys showed that several courses we relied on outside of CSE in our previous curriculum consistently scored low in these exit

surveys. To address this issue, we incorporated necessary material from these classes into our curriculum redesign.

A.4.6 Student course evaluations

During course evaluations, students both rate detailed aspects of the course and also give an overall score. These scores are one mechanism we use to monitor the quality of course delivery. While student happiness with a class is not a foolproof indicator of its quality, we have found a high correlation between the two.

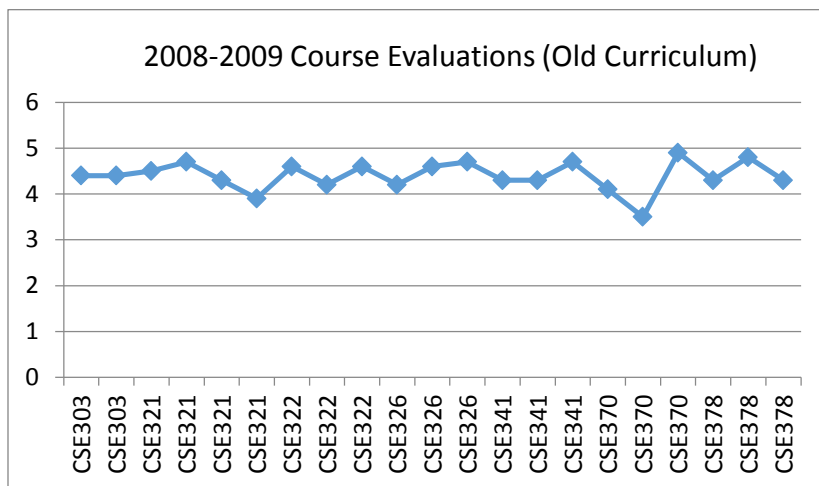


Figure 4-5 Prior Curriculum Course Evaluations

Student rankings are on a five point scale, but because the university applies a normalization function that takes into account the difficulty and workload of a course, “adjusted” scores can be somewhat higher than five. Figures 4-5 and 4-6 show the adjusted student ratings for each offering of each core course under our old and new curricula. The x-axis has each offering of each course taught in a particular academic year, and the y-axis is the adjusted mean overall score for that offering.

The 2008-2009 results reflect a mature curriculum that had undergone many rounds of improvement. The results are, with very rare exception, uniformly high. The 2011-12 results show much greater variability in scores. While the highest ratings for individual-offerings are achieved there, we also see a great deal of inconsistency, and some course offerings that might (and have) caused concern. Corrective actions have been taken, but clearly we are not yet where we would like to be in steady-state.

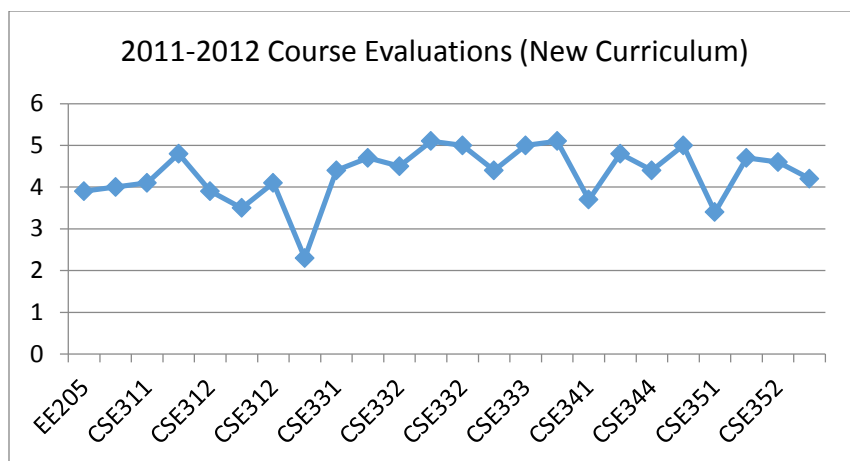


Figure 4-6 New Curriculum Course Evaluations

A.4.7 Faculty self-evaluations

Faculty members evaluate the courses they teach by filing a self-assessment each quarter with the Department Chair. This includes not only a post-mortem examination of the faculty member's effectiveness as a teacher but also comments regarding course pre-requisites, development, infrastructure, assignments, etc. Annually, two peers evaluate each and every faculty member, regardless of rank, for teaching performance and effectiveness. Self-assessments, student evaluations, and informal observations by other faculty who attended lectures are collected and overall advice developed to help the faculty member improve their performance.

A.4.8 Capstone design projects

The capstone courses are the centerpiece of our program, and are key to providing and assessing student outcomes. In a capstone course, students work in teams to design and implement a substantial project comprising multiple components. As part of their project, they produce project reports and make class presentations where they present their design, describe the design choices they made, and the tradeoffs they considered. At the end of the course, the project teams demonstrate their projects to their peers, and in many cases to members of affiliate companies and to the department as a whole.

Many outcomes are integral to the student's success in the capstone course, from the mastery and application of fundamental concepts in mathematics, science and computer engineering, to evaluating design tradeoffs and making effective design decisions, to working effectively as part of a team, to self-learning of new concepts and tools, to understanding the impact of computer engineering on society, to effective oral and written communication. The project reports, demonstrations and presentations provide the instructor a direct mechanism to assess whether students have met outcomes b through j. In addition, we often invite visitors from our affiliate and other companies to capstone course presentations and demonstrations, and ask them to help us verify that the projects are at the appropriate level and targeting relevant real-world problems. We also capture many projects from our capstone projects in video productions that highlight the projects as well as the design process and tools the students used in the class. This allows our outside constituencies, particularly our industrial affiliates, to assess the quality and scale of

these capstone projects. These videos are available online at www.cs.washington.edu/info/videos/.

A.4.9 Per-course Instructor Meetings

As part of the new curriculum roll-out, we have held academic year end review meetings of the instructors who have taught each of the new courses. The goals of these meetings have been to share opinions experiences, to discussion what is working and what is not, and to come to decisions about changes to be made. These meetings are also being used to refine a set of course syllabi that document many of the decisions. (These syllabi are available publically, at <http://www.cs.washington.edu/education/ABET/syllabi/static/>. Copies are included in Appendix A.) The meetings have been particularly useful for considering the details of course content, now that we have experience trying to teach that material in our 10-week quarter.

A.4.10 Full-Faculty Curriculum Meetings

Because the per-course faculty meetings are best suited to making improvements within individual courses, we are also holding annual faculty meetings to discuss the curriculum as a whole. These meetings allow us to communicate concerns about material coverage in pre-requisites, and to work out solutions that may involve changes to both the pre-requisite and the downstream course. These meetings also give us a chance to review the educational experience we provide as a whole. We find that having experts on all our classes in the same room at once is an effective way for each of us to gain additional context that can help with instruction of the individual courses we teach, and can enable efficient decision making about possible improvements. This ability to navigate quickly is crucial as we are still debugging our new curriculum.

B. Continuous Improvement

We discuss here changes to our program made in response to the results of past assessments, as well as future changes contemplated in light of the current results.

B.1 Targeted Assessment of Outcomes in Courses

a) an ability to apply knowledge of mathematics, science, and engineering

In 2007, the results for this outcome did not meet our standard of at least 80% of our students achieving at the Satisfactory level or greater. Partially in response to this, we redesigned and rolled out a new 300-level curriculum. That curriculum places greater emphasis on math, science, and engineering than our previous one. For instance, we now teach a probability course in house (CSE 312), and we have introduced engineering courses that had no counterparts in the earlier curriculum (CSE 331, CSE 333, and CSE 344). The most recent results indicate we are now attaining our standard.

b) an ability to design and conduct experiments, as well as to analyze and interpret data

We also failed to meet our standard on this outcome in 2007, but are meeting it in our most recent assessment. It was not one of the explicit objectives of the curriculum revision to address this outcome. However, the faculty was aware of the concern and may be placing greater emphasis on this outcome through small changes in individual courses.

c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

We failed to meet our standard on this outcome in 2007, although by such a small amount that it may well have been statistical variation. We have not taken specific steps to enhance this outcome, but current assessments show we are, if anything, doing slightly better, presumably as a side-effect of the curriculum revision.

d) an ability to function on multi-disciplinary teams

In 2007, we failed to meet our standard for this objective by a wide margin. Our most recent assessment shows we are now meeting our standard. This is an area where we took active steps to improve student achievement. First, we introduced CSE 331, Software Design and Implementation, to provide students with both abstract material and real experience working on a significant team project early in the program. Before the curriculum revision, that experience was available only in more advanced courses. Second, team project work has become more common in all of our courses, rather than being restricted to capstones and a few advanced courses, as it was in 2007. That is, an emphasis on teamwork has permeated the curriculum. Because students now typically engage in multiple team projects on the path to graduation, they are more likely to work in multiple, distinct roles, provided them with greater insight into teamwork. We believe the current assessment results show the effects of these changes.

e) an ability to identify, formulate, and solve engineering problems

The 2007 assessment results indicate we were meeting our standard on this outcome, as do the most recent assessment results.

f) an understanding of professional and ethical responsibility

Both the 2007 and most recent assessments indicate good achievement on this outcome. We did change our delivery of this material in the intervening years. In the past, it was taught as a separate track in what was essentially a tools course. With our update it has been integrated into three courses where it is a natural fit, and where its content is course-relevant.

g) an ability to communicate effectively

The 2007 assessment indicated we were not meeting our achievement standard for this outcome. In response, we made two changes:

1. In Spring 2010 we rolled out a new software engineering course, CSE 331. This course can be taken immediately on entry to the department. While the major topic is techniques for software construction, this involves significant team work, and consequently places significant emphasis on communication.
2. We increased the expectations for communication experiences in our capstones. Those courses now typically require regular, written design and implementation documents that

are maintained throughout the quarter; a public, oral presentation at the end of the quarter; and a written retrospective report.

The most recent assessment results are much stronger than the 2007 results.

h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context

The 2007 assessment indicated we were meeting our standard, and the most recent results indicate that if there has been any change, it has been towards improvement.

i) a recognition of the need for, and an ability to engage in life-long learning

The 2007 assessment results met our standard, but the most recent assessment results miss that standard by a small margin. Indirect feedback, in particular both our alumni surveys and employer comments, indicate that our graduates are strong on this outcome, so we do not feel that the current targeted assessment results indicate a crisis. It's notable that the individual course with the worst results among those assessing this outcome is the one that students are likely to take first among them. All assessed courses are at the 400-level. In general, our 400-level courses place increasing emphasis on students mastering topics, tools, and techniques on their own; we don't explain every detail to them. So, in that sense it's not surprising that the worst results come when the students have the least experience with this outcome.

To address this, we are considering introducing greater self-learning components in our 300-level courses, mainly by backing off on how detailed our instructions are for course projects. This sort of change requires review by the faculty curriculum committee, and then assuming it was approved, would be explained to the full faculty at our annual retreat meeting just before next autumn quarter begins.

j) knowledge of contemporary issues

The 2007 assessment met our standard, as do the most recent results. We have not made any changes specifically targeted to this outcome.

k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Both the 2007 and recent assessment results indicate we miss our standard by a small margin. For the most recent results, we have evaluation for only a single course offering at the time of this writing (but are expecting an additional Spring quarter course to complete the evaluation very shortly). These results indicate that we should consider taking some action to try to improve this outcome. Based on indirect measures (in particular, student feedback), we have chosen to strengthen student experience with source control. In particular, we are working to deploy a local github server, and to uniformly make use of it as a regular experience in our software oriented courses.

B.2 Additional Assessment Processes

We present here a summary of the changes that have resulted from our indirect assessment processes

New Courses

In addition to the core curriculum redesign, we have been very active introducing new courses. Many of these courses come from student requests, expressed directly to our advising staff and in quarterly meetings with the department Chair. Some are also reflections of our increasing faculty size and the consequent increase in diversity of areas of faculty interest. Table 4-6 presents summaries of these changes.

**Table 4-6 Recently Introduced Courses
(excludes core curriculum revision)**

<i>Course</i>	<i>Title</i>
<i>Pre-majors Courses</i>	
CSE 120	Computer Science Principles
CSE 131	Digital Photography
CSE 140	Data Programming
CSE 154	Web Programming
EE 205	Signal Conditioning for Computer Engineers
<i>Non-core Majors Courses</i>	
CSE 333	Systems Programming
CSE 344	Introduction to Data Management
CSE 390L	Leadership Seminar Series
CSE 446	Machine Learning
CSE 452	Distributed Computing
CSE 486	Synthetic Biology
CSE 487	Advanced Systems and Synthetic Biology
CSE 488	Laboratory Methods in Synthetic Biology
CSE 490E	Entrepreneurship
<i>Capstone Design Courses</i>	
CSE 441	Human-Computer Interaction Capstone
CSE 460	Computer Animation
CSE 481H	Accessibility Capstone
CSE 481I	Sound Capstone
CSE 481K	Technology for Resource-Constrained Environments
CSE 481M	Home Networking Capstone
CSE 481O	Kinect Capstone

Other Program Changes

We have also made numerous other changes to our program in response to feedback obtained through the indirect channels. Table 4-7 lists these.

Table 4-7 Additional Changes

<i>When</i>	<i>Concern</i>	<i>Source</i>	<i>Response</i>	<i>Results</i>
2007	Insufficient ethics content	Exit surveys	Faculty meet to discuss	Ethics component of CE curriculum embedded into CSE 403, 484 and 466.
2007-2008	Students felt keeping tools component from old CSE 303 was very important in new curriculum	Small feedback groups solicited during curriculum revision, Exit Surveys	Faculty met to determine how a new course might be organized	CSE 390A was instituted as an optional course for students to gain a stronger familiarity with Unix and other software tools.
2007-2009	CE too restrictive, want more flexibility	Exit surveys	Faculty meet to discuss	With new curriculum, create a CE degree that is more flexible and allows students to focus across the curriculum.
2007-2009	Longer capstone experiences	Exit Surveys, Faculty input	Faculty meet to discuss	
2007-2009	Students began requesting a machine learning class for undergrads	Exit surveys	AI and Machine learning faculty meet to develop new course	CSE 446 added to the curriculum
2009	CSE 401 sections needed	Faculty feedback and course evaluations	Faculty teaching 401 met to discuss	401 section added greatly enhancing student participation in the course
2009	Feedback from students wanting more experience with Web applications	Exit survey, faculty feedback, affiliate feedback	Lecturer Marty Stepp develops new course	CSE 154 is introduced into the curriculum. A major's version of this non-major course is added in Fall 2012 due to the high interest
2009	Request for a distributed systems course	Exit Surveys, Faculty feedback, Affiliate feedback	Faculty meet to develop new course	CSE 452: distributed systems introduced into the curriculum

When	Concern	Source	Response	Results
2009	Students didn't like the structure of CSE 303	Exit surveys, faculty feedback	Part of new curriculum development was to fix this problem	CSE 333 Systems Programming and 331 Software Implementation were used to enhance these two areas
2009-2010	Lack of appropriate EE background for computer engineering students	Faculty evaluation of student abilities, student self-assessment and comments.	Committee formed to discuss a new version of EE 215 fundamentals	Created EE 205, a specialized version of EE 215 designed to meet the needs of CE Majors
2010	Request for more robotics courses	Exit surveys	Faculty meet to discuss	Several capstone and project courses are added to the curriculum focused on robotics and 3D camera development.
2011	Concerns about scheduling of courses – too many in winter, not enough in spring	Exit Surveys	Undergraduate advising team and Associate Chair meet	Try to better balance courses for the following year to make sure electives are not clustered too heavily into one course.
2007-2011	Complaints about department website	Exit surveys, Affiliate feedback	Chair, Staff and Faculty meet to create a new website	New CSE website is designed and deployed
2011	Database experiences could be more in-depth	Exit surveys, Faculty feedback	Database faculty discuss	Create a CSE 344 introductory databases course with a follow-on CSE 444 Database Internals course where students can now gain greater exposure to the field
2009-2012	Request more exposure to entrepreneurship process	Exit surveys, Advisor feedback	Faculty discuss with College	CSE 490 Entrepreneurship is started in 2011 and offered as part of an Entrepreneurship program run by the Dean of the College of Engineering. Students are also encourage to participate in Business plan competitions at the Business School. (This course is now CSE 390L.)
2009-2012	Want more HCI	Exit surveys	Faculty discuss	CSE 440 HCI is taught twice a year now, and we added an HCI capstone course (CSE 441) as well.
2007-2012	Complaints about technical writing courses	Exit surveys, in person meetings	Faculty and advisors meet to discuss	Pilot a CSE 490W scientific writing course for current Computer Engineering and Combined BS/MS students. Very positive feedback from the instructor. With the college disbanding HCDE 333, this could be a good alternative when time allows for this course in the schedule.

When	Concern	Source	Response	Results
2010-2012	Students want more opportunity to pursue research	Exit survey, in person meetings with students	Advisors and students and faculty discuss	In Winter 2012, the students with advisor support start their own research group seminar to discuss current research areas. This will continue in spring 2013.
2012	Some 400 level courses need 333 as a prereq	Full faculty meeting feedback, Exit surveys, Affiliates feedback	Faculty convene a meeting to discuss	CSE 333 is added as a prerequisite course requirement to CSE 451 (Operating Systems), 457 (Graphics), and 461 (Networks)
2012	CSE 312 contained too much material for one quarter, and wasn't sufficiently coherent	Full faculty meeting feedback, student surveys	Removed computation complexity topic from CSE 312	CSE 312 now focuses exclusively on prob/stat; computational complexity is finding a new home

Currently Contemplated Program Changes

In addition, we are currently evaluating additional changes, motivated primarily by the indirect feedback channels.

As one example, we held our latest full faculty curriculum review meeting in Autumn 2012. An idea that came from it, which we are studying for potential implementation, is to uniformly administer “first day quizzes” in our courses. The purpose of these quizzes would be to assess student fluency in the prerequisite material by asking a small number of questions on the most relevant topics of the prerequisite courses. A number of faculty had already been doing this routinely and found the quizzes were useful for gauging the preparation of the particular groups of students comprising the class, were useful to the students as a kind of review, and also helped convey to the students that courses build upon each other. As a department, we are also interested in the possibility of using these first day quizzes as an assessment tool, evaluating the effectiveness of our pre-requisite courses by tracking how students do in these downstream first day quizzes.

As another example, experience with student writing by faculty teaching capstones has led to a proposal that we offer university writing course credit (<http://www.washington.edu/uaa/advising/degreeplanning/writreqs.php>) for each of our capstone design courses. That option requires that students prepare a 10 to 15 page paper, that the paper be reviewed and feedback provided, and that the student then revise the paper based on the feedback and turn in a revision. Additionally, the suggestion is that the paper's topic be an examination of ethical issues in the capstone's domain, and a summary of the arguments on each side of each issue. Some of our experience with student writing indicates that they are much more proficient at informal prose than at technical writing, whose primary goal is to convey information clearly and concisely. The capstone paper experience would be oriented to the latter.

CRITERION 5. CURRICULUM

Since our last ABET program review in 2007, the department has designed and implemented an update to our core curriculum, replacing all core 300 level courses in it with a set of new courses. These changes were motivated by and designed with feedback from undergraduates and employers, as well as the observations of our own faculty. We began teaching the new curriculum in Spring 2010. During a transition period we offered both the old and new courses. We completed introducing the new set of courses during the 2010-11 academic year, and at this point are no longer offering the old courses. We are just now seeing the graduation of the last of the students who started the program under the old set of courses and graduation requirements.

The summary of the curriculum redesign is that we moved from a traditional “stovepipe” model where each course focused on one set of concepts, to a model where concepts from different areas are presented together to highlight the interaction and interplay among these core concepts. For example, we now teach a course – CSE351: The Hardware/Software Interface – that introduces topics from programming languages, compilers, computer architecture and operating systems as a coherent set of ideas.

This revamping of the curriculum has also allowed us to modify the graduation requirements to make them substantially more flexible. This reflects the changing nature of our field, which has seen an explosion in new applications of computer engineering and opportunities for collaboration and impact in other fields, such as the bio-sciences, environmental sciences, global health, and economic development.

We describe here the motivation for the revision, the process by which the revision was carried out, and the curriculum that resulted from the revision. Our assessment processes for this revision are described in Section 4, Continuous Improvement.

Motivation and Goals

We were motivated to update our core curriculum both because of feedback from students and employers and because of our own evaluation of the state of the then existing courses. Our existing curriculum was originally composed in an era where single-core, desktop computing was the predominant model, and while our curriculum had evolved, we needed a careful examination of what balance of material would best serve our students in the current and near future world. One major goal of the revision was to modernize the course content.

We had also noticed that both employers and students were interested in an ever increasing diversity of knowledge and skill sets, and were aware that other universities had undertaken curriculum revisions where flexibility was the paramount goal. Our existing curriculum grouped all computer engineering students into one of two bins: hardware-oriented or software-oriented. It became clear that this was not sufficient, that there were many career paths within computer engineering that a student could pursue, and that the best mix of courses to provide the knowledge and skills required depended on the path chosen. We also felt that defining a few additional fixed paths was not sufficient, as the increasing diversity of paths would continue. Our goal was to accommodate this growing diversity by increasing flexibility, primarily by defining a

more compact core. Our own external advisory committee also recommended that we follow that course.

Curriculum Update Process

An initial effort to redefine the curriculum began in summer of 2008. A group of about eight faculty, representing a broad range of areas within the department, met throughout the summer to work out the basic design. They adopted a clean slate approach, and eventually presented a proposal that affected all of our courses, from pre-major up through advanced, at our annual faculty meeting in the fall. The initial faculty response was positive – we liked the idea of redesigning the curriculum – but in short order the practical difficulty of making such sweeping changes became overwhelming, and we decided to start again, this time focusing only our 300-level courses, which comprise the required core of our degree requirements.

In January 2009 the department began a systematic revision of the undergraduate core curriculum and requirements. In February, the goals and status of this revision were presented to our External Advisory Committee (EAC), consisting of Barbara Liskov (MIT, chair), Randy Bryant (CMU), Deborah Estrin (UCLA/Cornell), Tom Mitchel (CMU), Christos Papadimitriou (Berkeley), Pat Hanrahan (Stanford), Eric Horvitz (Microsoft Research), and Jeff Dean (Google). The EAC strongly recommended a modernization of our undergraduate curriculum.

In late May, the faculty overwhelmingly approved the proposed revision "in principal", with work on details, degree requirements, etc. to be accomplished during the Summer of 2009. Following that, a more polished, specific proposal was presented for comments to both our EAC and our undergraduates. The response from the EAC was positive; with the only concern voiced being that the proposal might be optimistic about the amount of material that could be covered in a one quarter course. The undergraduates were positive as well, but had some concerns about the interface between the new core courses and the senior-level courses that followed, motivated primarily by very practical concerns (in particular, the amount of C programming experience the new curriculum would afford). The details of the proposal were modified in response to this feedback, and formal faculty approval of the final revision took place in Autumn 2009. The first offering of the first new course took place in Spring 2010, and all new courses were rolled out by the end of the 2010-11 academic year.

During that period we had students who had started our previous curriculum working towards our old graduation requirements, while around them we were reducing offerings of those courses and moving toward a completely different set of requirements. For that reason, an unusually flexible set of graduation requirements was instituted for “transitional students” – those who had begun taking core majors courses in the three quarters from Summer 2009 through Winter 2010. Those transitional requirements can be seen at <http://www.cs.washington.edu/students/ugrad/transition/>. Students who started the program earlier than Summer 2009 were expected to graduate under the old requirements; those starting in Spring 2010 or later were expected to graduate under the new requirements. Old, new and transition requirements were designed to contain all necessary elements of our ABET accredited program.

Effect of the Revision

We have had three sets of graduation requirements for our computer engineering degree since Autumn of 2007, not counting the transitional requirements put in place temporarily for a limited

group of students. We try to give here an overall sense of the changes that have taken place. Details on the graduation requirements can be found online, however:

Autumn 2007 – Spring 2009	http://www.cs.washington.edu/education/ugrad.old/current/CE_Aut07_Spr09.pdf
Summer 2009 – Winter 2010	http://www.cs.washington.edu/students/ugrad/transition/
Spring 2010 – Spring 2011 (Transitional)	Hardware Specialization: http://www.cs.washington.edu/public_files/ugrad/curriculum/CEHW_reqs_wi11.pdf Software Specialization: http://www.cs.washington.edu/public_files/ugrad/curriculum/CESW_reqs_wi11.pdf
Autumn 2011 - present	http://www.cs.washington.edu/public_files/ugrad/curriculum/CE_reqs_au11.pdf

At the highest level, our degree requirements consist of a “Computer Engineering Component” and a “General Education Component.” The total number of credits required in each of those two broad categories has remained nearly constant – there has been no significant reduction or rebalancing. Within the general education component, the “Mathematics & Natural Sciences” was solidified to ensure all students complete 45 credits. A concern from our 2007 ABET review mentioned that students *could* complete the Computer Engineering degree with slightly under 45 math/science credits. When we decided on the new requirements, we specifically took this concern into account to make certain that students are guaranteed to meet this requirement. We note however that many of our students complete the math minor at UW, so students generally take more than 45 credits.

Within the Computer Engineering component we distinguish between Required courses and what we will call here Selective Electives. The former are required; there is some flexibility in the latter (along the lines of “4 courses chosen from...”), but the intention is to ensure that every graduate has taken a set of courses that together meet our student outcomes.

Over this six year evolution in requirements, the important trend is a reduction in the number of required credits, from 47 in 2007 to 33 today, and an increase in the number of selective elective credits, from 27 in 2007 to 39 today. It is this change that affords our curriculum the flexibility to embrace the “Pathways” approach described in the next section.

The 8 credit reduction in the required course component of our curriculum is due to moving two courses from the Required to the Selective Elective category. Those two courses are computer networking and operating systems. It was felt that they were required primarily for historical reasons – they were the only courses that met some broad educational goals at the time we composed the curriculum – but with the introduction of new 400-level courses over time, that was no longer the case. Further, depending on the career direction a student intended to pursue, other mixes of 400-level courses might provide stronger preparation. Instead of specifically requiring the operating systems and networking courses of every graduate, we now put them in an equivalence class with Compilers, Software Engineering, Database Systems Internals, Embedded System Software, Advanced Digital Design, Computer Design and Organization, and Computer Security. Students are required to take four courses from that group. A further

constraint is imposed by subdividing that group and requiring students to take at least one course from each of two subcategories.

A final effect of the revision is that we are now teaching our own probability course, CSE 312. We made this change because of the increasing importance of that material throughout our discipline, and because previous sources of instruction were not proving effective for our students.

A. Program Curriculum

A.1 Tables

By far the most significant curricular activity over the past seven years has been the revision of our core courses. As an aid in understanding this section, we provide a specific list of old courses, which have been eliminated by the revision, and new courses, which have been introduced. We include among the eliminated courses a Probability/Statistics course taught by Math and a technical writing course offered by Human-Centered Design & Engineering. The latter is in fact being eliminated; the former will continue on, but we are now providing our own replacement course within our department, and so are no longer relying on the Math version.

Table 5-1 Core Curriculum Update Summary

Course	Title / Textbook
Old Curriculum Courses	
HCDE 333	Advanced Technical Writing and Oral Presentation
STAT 390	Probability and Statistics in Engineering and Science, <i>Applied Statistics for Engineers & Scientists</i> , Devore.
CSE 303	Concepts and Tools for Software Development, <i>C: A Reference Manual</i> , Harbison and Steele.
CSE 321	Discrete Structures, <i>Discrete Mathematics and Its Applications</i> , Rosen.
CSE 322	Introduction to Formal Models, <i>Introduction to the Theory of Computation</i> , Sipser.
CSE 326	Data Structures, <i>Data Structures & Their Algorithms</i> , Lewis and Denenberg.
CSE 370	Introduction to Digital Design, <i>Contemporary Logic Design</i> , Katz and Borriello.
CSE 378	Machine Organization and Assembly Language, <i>Computer Organization and Design</i> , Patterson and Hennessy.
New Curriculum Courses	
CSE 311	Foundations of Computing I
CSE 312	Foundations of Computing II, <i>Introduction to Probability</i> , Bertsekas and Tsitsiklis.
CSE 331	Software Design & Implementation, <i>The Pragmatic Programmer</i> , Hunt and Thomas.
CSE 332	Data Abstractions, <i>Data Structures and Algorithm Analysis</i> , Weiss.
CSE 333	Systems Programming, <i>C++ Primer</i> , Lippman, Lajoie & Moo.
CSE 351	The Hardware/Software Interface, <i>Computer Systems</i> , Bryant.
CSE 352	Hardware Design and Implementation
CSE 490W	Scientific and Technical Writing

Table 5-2 Curriculum
Computer Engineering

Year; Semester or Quarter	Course (Department, Number, Title)	Required, Elective or Selected (R, an E or SE.) ¹	Category (Credit Hours)				Last Two Terms the Course was Offered	Maximum Section Enrollment for the Last Two Terms
			Math & Basic Sciences	Eng Topics <i>Check if Contains Significant Design</i>	General Educatio n	Other		
1 Yr/ 1 st Qtr	Math 124 Calc w/Anal Geometry	R.	5	()			Wi 2013 & Sp 2013	120, 120
	Natural Science Elective	S.E.	5	()			Wi 2013 & Sp 2013	336, 510
	English Composition	S.E.		()	5		Wi 2013 & Sp 2013	22, 22
1 Yr/ 2 nd Qtr	Math 125 Calc w/Anal Geometry	R.	5	()			Wi 2013 & Wp 2013	120, 120
	Phys 121/131 Mechanics	R.	5	()			Wi 2013 & Sp 2013	220, 220
	VLPA/I&S Elective	S.E.		()	5		Wi 2013 & Sp 2013	Varied
1 Yr/ 3 rd Qtr	Math 126 Calc w/Anal Geometry	R.	5	()			Wi 2013 & Sp 2013	160, 160
	Physics 122/132 Electromagnetism	R.	5	()			Wi 2013 & Sp 2013	198, 176
	VLPA/I&S Elective	S.E.		()	5		Wi 2013 & Sp 2013	Varied
2 Yr/ 1 st Qtr	Natural Science Elective	S.E.	5	()			Wi 2013 & Sp 2013	130, 220

Year; Semester or Quarter	Course (Department, Number, Title)	Required, Elective or Selected (R, an E or SE.) ¹	Category (Credit Hours)				Last Two Terms the Course was Offered	Maximum Section Enrollment for the Last Two Terms
			Math & Basic Sciences	Eng Topics <i>Check if Contains Significant Design</i>	General Educatio n	Other		
	CSE 142 Computer Programming I	R.		4 (✓)			Wi 2013 & Sp 2013	345, 475
	VLPA/I&S Elective	S.E.		()	5		Wi 2013 & Sp 2013	Varied
2 Yr/ 2 nd Qtr	Math 308 Linear Algebra	R.	3	()			Wi 2013 & Sp 2013	50, 50
	CSE 143 Computer Programming II	R.		5 (✓)			Wi 2013 & Sp 2013	335, 335
	VLPA/I&S Elective	S.E.		()	5		Wi 2013 & Sp 2013	Varied
	Math Science from Approved List	S.E.	3	()			Wi 2013 & Sp 2013	Varied
2 Yr/ 3 rd Qtr	EE 205 Introduction to Signal Conditioning/ EE 215 option for either one	R.		4 (✓)			Wi 2012 & Wi 2013	25, 25
	HCDE 231 Intro. to Technical Writing	R.		()	3		Wi 2013 & Sp 2013	30, 30
	VLPA/I&S Elective	S.E.		()	5		Wi 2013 & Wi 2013	Varied
	CSE 311 Foundations of Computing II	R.	2	2 ()			Wi 2013 & Sp 2013	55, 82
3 Yr/ 1 st Qtr	CSE 351 The Hardware/Software Interface	R.		4 (✓)			Wi 2013 & Sp 2013	82, 90
	CSE 332 Data Structures	R.		4 (✓)			Wi 2013 &	115, 55

Year; Semester or Quarter	Course (Department, Number, Title)	Required, Elective or Selected (R, an E or SE.) ¹	Category (Credit Hours)				Last Two Terms the Course was Offered	Maximum Section Enrollment for the Last Two Terms
			Math & Basic Sciences	Eng Topics <i>Check if Contains Significant Design</i>	General Educatio n	Other		
							Sp 2013	
	VLPA/I&S Elective	S.E.		()	5		Wi 2013 & Sp 2013	Varied
3 Yr/ 2 nd Qtr	CSE 312 Foundations Computing II	R.	2	2 ()			Wi 2013 & Sp 2013	90, 55
	HCDE 333 Adv. Tech. Writing & Oral Presentation (or approved UW Writing course Fall 2013)	R. (S.E.)		()	4		Wi 2013 & Sp 2013	30, 30
	Free Electives	E.				2	Wi 2013 & Sp 2013	Varied
	CSE 352 Hardware Design & Implementation	R.		4 (✓)			Au 2012 & Sp 2013	30, 50
3 Yr/ 3 rd Qtr	CSE Core Course	S.E.		4 (✓)			Wi 2013 & Sp 2013	55-75
	CSE Core Course	S.E.		4 (✓)			Wi 2013 & Sp 2013	55-75
	Free Elective	E.		()		7	Wi 2013 & Sp 2013	Varied
4 Yr/ 1 st Qtr	One of CSE401, CSE403, CSE 466 or CSE 484	S.E.		4 (✓)			Wi 2013 & Sp 2013	55-75
	One of CSE401, CSE403, (16) CSE444, CSE451, CSE461, CSE466, CSE467, CSE471, and CSE484;	S.E.		4 (✓)			Wi 2013 & Sp 2013	55-75
	Free Elective	E.		()		6	Wi 2013 &	Varied

Year; Semester or Quarter	Course (Department, Number, Title)	Required, Elective or Selected (R, an E or SE.) ¹	Category (Credit Hours)				Last Two Terms the Course was Offered	Maximum Section Enrollment for the Last Two Terms
			Math & Basic Sciences	Eng Topics <i>Check if Contains Significant Design</i>	General Educatio n	Other		
							Sp 2013	
4 Yr/ 2 nd Qtr	One of CSE401, CSE403, (16) CSE444, CSE451, CSE461, CSE466, CSE467, CSE471, and CSE484;	S.E.		4 (✓)			Wi 2013 & Sp 2013	55-75
	One of CSE401, CSE403, (16) CSE444, CSE451, CSE461, CSE466, CSE467, CSE471, and CSE484;	S.E.		4 (✓)			Wi 2013 & Sp 2013	55-75
	Free Elective	E.		()		4	Wi 2013 & Sp 2013	Varied
	Free Elective	E.		()		4	Wi 2013 & Sp 2013	Varied
4 Yr/ 3 rd Qtr	Capstone Course	S.E.		5 (✓)			Wi 2013 & Sp 2013	25
	CSE Senior Elective	S.E.		5 (✓)			Wi 2013 & Sp 2013	55-75
	CSE Senior Elective	S.E.		5 (✓)			Wi 2013 & Sp 2013	55-75
	Free Elective	E.		()		2	Wi 2013 & Sp 2013	Varied
			45	68	42	25		
TOTALS-ABET BASIC-LEVEL REQUIREMENTS			45	67.5				
OVERALL TOTAL CREDIT HOURS FOR COMPLETION OF THE PROGRAM			45	68	42	25		
PERCENT OF TOTAL			25%	38%				

Year; Semester or Quarter	Course (Department, Number, Title)	Required, Elective or Selected (R, an E or SE.) ¹	Category (Credit Hours)				Last Two Terms the Course was Offered	Maximum Section Enrollment for the Last Two Terms
			Math & Basic Sciences	Eng Topics <i>Check if Contains Significant Design</i>	General Education	Other		
Total must satisfy either credit hours or percentage	Minimum Semester Credit Hours		32 Hours	48 Hours				
	Minimum Percentage		25%	37.50%				

Required courses are required of all students in the program, **elective** courses (often referred to as open or free electives) are optional for students, and **selected elective** courses are those for which students must take one or more courses from a specified group.

1. For courses that include multiple elements (lecture, laboratory, recitation, etc.), indicate the maximum enrollment in each element. For selected elective courses, indicate the maximum enrollment for each option.

A.2 Curriculum Alignment with Program Educational Objectives

Our program educational objectives are engineering quality, leadership, economic impact, and lifelong learning. All of these require strong preparation in the concepts, skills, and abilities of a computer engineer. Additionally, they require an understanding of the context in which contributions may be made and the processes by which substantial projects can be accomplished. Our curriculum aims to provide experiences that enable our students to perform at a high level in all these areas.

Our computer engineering curriculum provides depth and breadth across topics in computer science and computer engineering as described in detail in Section B.4 and Appendix A. The specific required mathematics, science and engineering knowledge is acquired as follows:

- Discrete mathematics is covered in depth in CSE 311 (Foundations I) and extended in other courses such as CSE 332 (Data Abstractions) and CSE 352 (Hardware Design and Implementation) which makes heavy application of Boolean algebra.
- Probability and Statistics is taught in CSE 312.
- Differential and integral calculus is covered in MATH 124, 125 and 126, the sequence on calculus and analytic geometry.
- Science background is covered in Physics 121 (Mechanics) and Physics 122 (Electromagnetism and Oscillatory Motion) and the natural science requirement.

Our computer engineering program focuses on the interface in computer systems between the hardware and software level. That is, computer engineers are faced with the design of systems that involve significant interaction between hardware and software and must have the knowledge required to deal with both sides of this interaction. Traditionally, this interface has been drawn close to the circuit level of hardware, where the computer engineer must have knowledge of analog and digital circuits, and understand computing systems comprised of these elemental building blocks. All of our students must take CSE 351, the Hardware/Software Interface, introduced as part of recent curriculum revision squarely focuses on this topic.

Students in computer engineering learn the underlying fundamentals of circuits and electrical engineering in EE 205 (Introduction to Signal Conditioning) or EE 215 (Introduction to Electrical Engineering), and the fundamentals of digital circuits in CSE 352 (Hardware Design and Implementation). Both courses involve a substantial laboratory component and hands-on design.

All students must take at least one of CSE 403 (Software Engineering), CSE 466 (Software for Embedded Systems), or CSE 484 (Security). These courses involve team work, a consideration of ethical responsibility, and an awareness of contemporary issues. They also reinforce earlier lessons about the hardware/software interface.

All students must take at least one capstone design course. These courses address a large variety of specific topics, but all include heavy emphasis on team work, design, independent learning, and communication. The typical format involves a combination of a large project design and implementation and some number of lectures. Project goals are typically set by the team, within a broad set of constraints imposed by the instructor. Regular assessment of progress is performed, and most often teams are required to regularly document the current state of the

project and to provide a plan for its completion. Public presentations of projects are commonplace, although sometimes substantial reports take their place.

In summary, students in the computer engineering program learn the concepts required to analyze and design complex systems comprising hardware and software at the level appropriate to their specialization.

A.3 Curriculum & Student Outcomes

Table 5-3 – Student outcomes and mapping to curricular and extra-curricular elements.

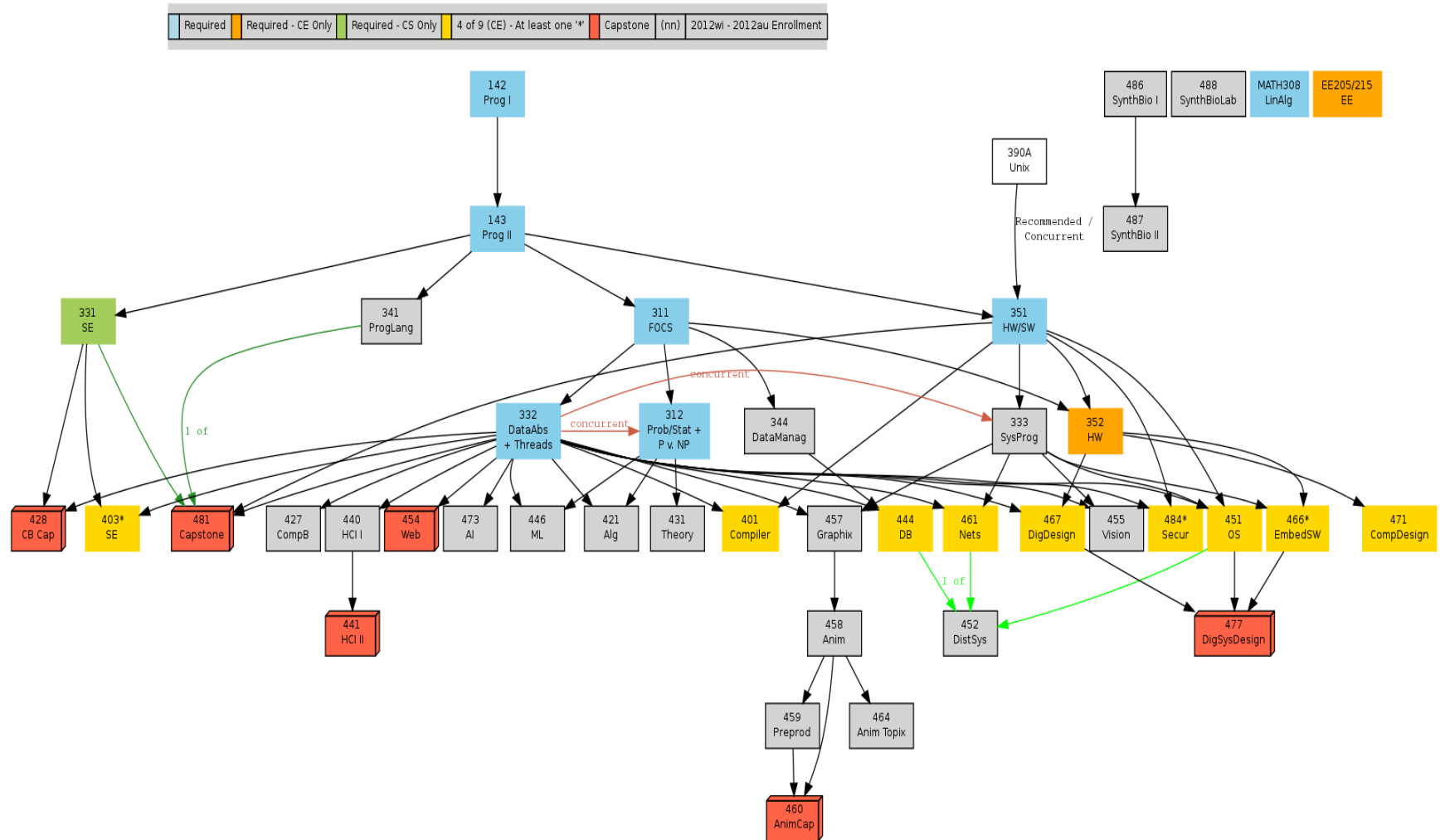
	Curriculum Component (credits)								Extra-curricular		
	VLPA/IS (30)	Commun (12)	Math/ Statis (18-24)	Science (20)	CE core (33)	Selective Elective (39)	Capstone	Electives	Industry internships	Research project	Teaching assistantships
(a) an ability to apply knowledge of mathematics, science, and engineering			M	M	M	H	H	H			
(b) an ability to design and conduct experiments, as well as to analyze and interpret data				M	M	H	H	H		M	
(c) an ability to design a system, component, or process to meet desired needs within realistic constraints					M	H	H	H	H	M	
(d) an ability to function on multi-disciplinary teams						L	H		H	M	
(e) an ability to identify, formulate, and solve engineering problems					M	H	H	H	M	H	
(f) an understanding of professional and ethical responsibility					M		M		M		
(g) an ability to communicate effectively		H			M	L	H		H	M	M
(h) the broad education necessary to understand the impact of engineering solutions	H						M		M		
(i) a recognition of the need for, and an ability to engage in life-long learning						M	H	M	H	H	
(j) knowledge of contemporary issues	M				M	M	M		H	H	
(k) an ability to use the techniques, skills, and modern engineering tools					H	H	H	H	H	H	

Table 5-4 Student outcomes and mapping to curriculum requirements.

Outcome		a	b	c	d	e	f	g	h	i	j	k
Core Courses												
CSE142	Computer Prog I	M				M						H
CSE143	Computer Prog II	M				M						H
CSE311	Foundations I	H				M						
CSE312	Foundations II	H				M						
CSE332	Data Abstractions	M	M	M		M						M
CSE351	HW/SW Interface	M	H	M		M	M			M		H
CSE352	HW Design and Impl	H	M	H		M				M		H
EE205 / EE215	Fund of Electrical Engineering	H	M	M	M	M		M				M
4 of 9												
CSE401	Compilers	H		M		H						H
CSE444	Databases	H	H	H		H				M		H
CSE451	Operating Systems	H	H	H		H				M		H
CSE461	Networks	H	H	H		H				M		H
CSE467	Advanced Digital Design	H	H	H		H				M		H
CSE471	Computer Design	H	H	H		H				M		H
<i>Extended core (at least 1)</i>												
CSE403	Software Engineering	H	H	H	M	H	H	M	M	M	H	
CSE466	Embedded Systems	H	H	H	H	H	H	H	H	M	M	
CSE484	Security	H	H	H	M	H	H		M	M	H	
<i>Capstone Design (at least 1)</i>												
CSE428	Comp Bio Capstone	H	H	H	H	H	M	H	M	H	M	H
CSE454	Internet and Web Serv		H	H	H	M		H			M	M
CSE460	Animation Capstone	H	H	H	H	H	M	H	M	H	M	H
CSE477	Digital System Design	H	H	H	H	H	M	H	M	H	M	H
CSE481A	Op Sys Capstone	H	H	H	H	H	M	H	M	H	M	H
CSE481C	Robotics Capstone	H	H	H	H	H	M	H	M	H	M	H
CSE481D	Games Capstone	H	H	H	H	H	M	H	M	H	M	H
CSE481H	Accessibility Capstone	H	H	H	H	H	M	H	M	H	M	H
CSE481I	Sound Capstone	H	H	H	H	H	M	H	M	H	M	H
CSE481K	Resource-Constrained Environ Capstone	H	H	H	H	H	M	H	M	H	M	H
CSE481M	Home Nets Capstone	H	H	H	H	H	M	H	M	H	M	H
CSE481O	Kinect Capstone	H	H	H	H	H	M	H	M	H	M	H

Outcome		a	b	c	d	e	f	g	h	i	j	k
Senior Electives - Also any courses listed above												
CSE331	Software Design and Implementation	H		M		H						H
CSE333	System Software	M		M		H						H
CSE341	Programming Languages	H		M		M				M		H
CSE344	Intro to Databases	H		M							M	H
CSE421	Intro to Algorithms	H										
CSE427	Computational Biology	M	H	M							M	
CSE431	Intro to Theory of Computation	H										
CSE440	Intro to HCI	M	H	H	H	H	H	H	M	M	M	H
CSE446	Machine Learning	M	H	H	M	H	M	M	M	M	M	M
CSE455	Computer Vision	H	M	M		M				M		M
CSE457	Computer Graphics	H		M	M	M			M	M		H
CSE473	Intro to AI	H	H	H		M				M	M	M
CSE 486	Intro to Synthetic Biology	H		H		H						
CSE 487	Adv Sys & Synth Bio	H		H								
CSE 488	Lab Methods in Synth Bio	H	H	H	H		H		H		H	H
CSE490W	Scientific and Technical Writing	H					M	H	H	H	H	

A.4 Course Prerequisite Structure



A.5 Meeting Degree Requirements

The curriculum is structured to provide both depth and breadth. Students majoring in Computer Engineering have the same flexibility to tailor the program toward different career specializations. For example, we offer capstone design courses in topics as diverse as robotics, computational biology, embedded systems, and home networking. We do not require any single such course; instead we ensure that all of them fulfill broad curriculum needs, such as training in communication and extensive design and team work, and then allow students to choose.

A set of common core courses form a foundation on which senior requirements are layered and appropriate capstone design courses provide an in-depth design experience. Our courses are structured to ensure that all students can achieve all the outcomes that we have defined for the program.

Our curriculum requirements are grouped into 9 categories:

- General education
- Visual, literary, and performing arts/Individuals & societies
- Written and oral communication
- Mathematics and statistics
- Science
- CSE core
- CSE senior electives
- Free electives

General education and VLPA/IS¹ requirements expose our students to a variety of topics in the humanities and the natural world. Students select the courses they wish to take in these areas based on their interests, and we encourage them to explore general education outside of their chosen degree/career path. General education courses complement the technical aspects of our computer engineering degree and support our program objectives by expanding our students' experiences and worldview. The field of computer engineering is highly interdisciplinary, and a student's ability to communicate ideas, work in teams, obtain new knowledge, and understand others is necessary for technical skills to be useful. A broad perspective and diverse skill base are fundamental to our students' ability to effectively engage in engineering quality, leadership, economic impact, and lifelong learning across interdisciplinary environments.

For the various pathways students follow, we suggest specific course types to meet these requirements (<http://www.cs.washington.edu/students/ugrad/pathways/>). For example in Human Computer Interaction, we encourage art and design courses; for a student wanting to go into the software industry, we suggest communication and business courses.

¹ Visual, Literary, and Performing Arts / Individual and Societies

Written and oral communication requirements include two technical writing courses² in addition to their required English composition course. Mathematics includes one-year of calculus and a linear algebra class. (We now teach a required probability/statistics class as CSE 312.) The science requirement includes two courses of Physics and 10 more credits of approved Natural Science. Students further increase their range by taking an additional three to six credits of approved natural science or higher-level math/statistics to bring their credit total to 45. They also all take CSE 311 and 312 where we count two credits each towards math. Free electives provide opportunities for students to expand their learning in any given area. A number of our students pursue a foreign language, a minor, or a double degree using this mechanism. Many use these credits for additional research credit in the department (6 credits of which can be counted in CSE electives, 9 credits if part of the Department or College Honors program) or to take more electives within the department.

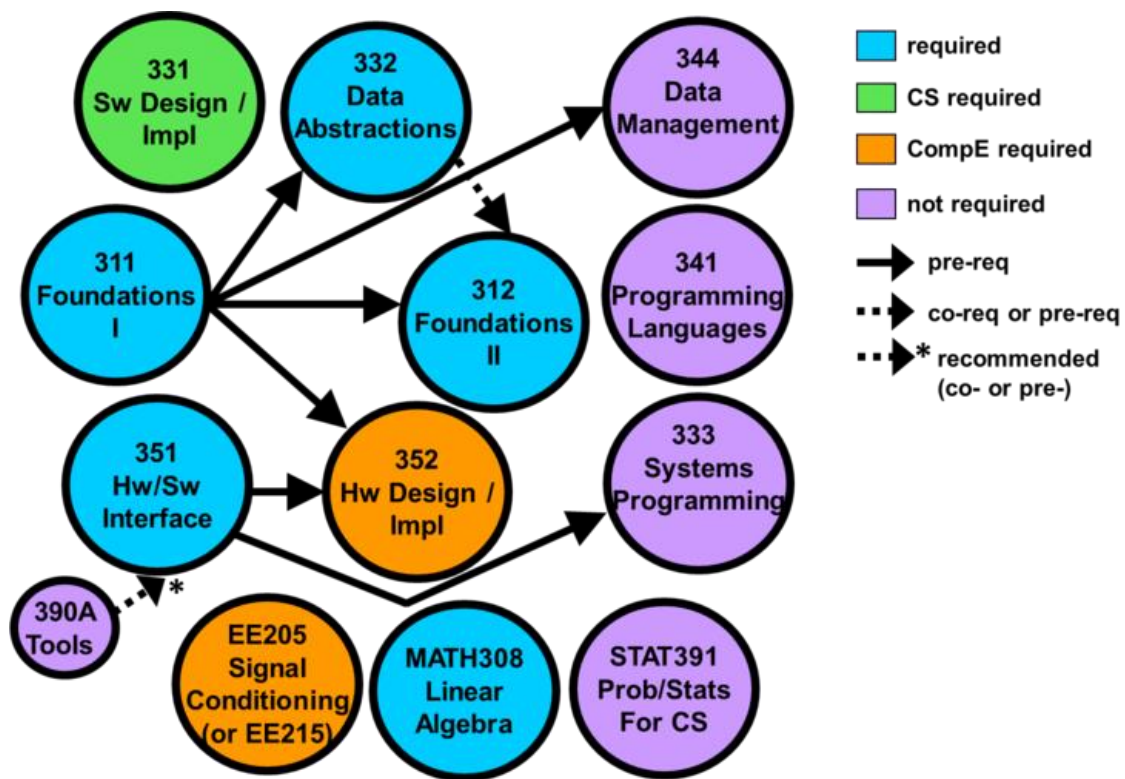


Figure 5-1 – Required Course Pre-requisite Structure

Figure 5-1 shows the pre-requisite structure of our 300-level courses. The heart of our curricular requirements is the departmental core requirements which are arranged in three tiers. First,

² The manner in which training in writing will be provided is being reviewed, as the College is terminating one of the writing courses we have relied on. We have already experimentally offered an in-department replacement, CSE 490W.

students in both Computer Engineering and Computer Science take common core requirements that ensure a solid foundation in the discipline. This component consists of the same 4 courses for both majors (CSE 311, 312, 332, and 351). Computer engineering students also take 2 additional foundation courses: EE 205 or 215 (Introduction to Electrical Engineering or Introduction to Signal Conditioning) and CSE 352 (Hardware Design & Implementation).

The list of CSE core courses can be found at:

http://www.cs.washington.edu/students/ugrad/core_courses/.

Senior electives can be found at: http://www.cs.washington.edu/students/ugrad/cse_electives/.

The Computer Engineering curriculum is described in detail on the department undergraduate program site http://www.cs.washington.edu/students/ugrad/degree_requirements/.

In summary, the program consists of the following requirements based on 180 credits taken under a quarter system:

Core requirements (33 credits): CSE 142, 143, 311, 312, 332, 351, 352, and EE 215/205.

CE Senior Electives (39 credits)

- Four courses chosen from CSE 401, 403, 444, 451, 461, 466, 467, 471, and 484; at least one of which must be CSE 403, 466, or 484.
- Two additional courses from the CSE Core Course list on the CSE site: http://www.cs.washington.edu/students/ugrad/core_courses/.
- One design capstone course from the approved list on the CSE site: <http://www.cs.washington.edu/students/ugrad/capstone/>
- 10-12 additional credits from the CSE Electives list, including at least 7 credits from College of Engineering courses, to bring the total to 39 credits.

Math and Natural Sciences (45 credits)

- MATH (18 credits): Math124, 125, 126 and 308
- Science (20 credits): PHYS 121, 122; 10 credits of Natural Sciences
- CSE 311 (2 credits apply to math)
- CSE 312 (2 credits apply to math)
- Additional Math/Science Credits (3-6 to bring total to 45): Courses chosen from the approved natural science list on the CSE site: http://www.cs.washington.edu/students/ugrad/natural_science/ or Stat 390, 391, 394, Math 307, 309, 334, 335, AMATH 351, 353. (Stat 391 recommended.)

VLPA and I&S (30 credits): from UW provided list.

Written and Oral Communication (12 credits): English Composition, HCDE 231, 333

Free Electives (20-25 credits; depending on courses chosen)

A.6 Design Experience

Students gain significant design experience throughout the program, starting with relatively small design problems in the core courses and proceeding to substantial, quarter-long team projects in the capstone courses. Many of the required senior-level courses are taught in the context of a substantial design project experience.

Selective Elective Courses

These courses all contain a serious design component.

CSE401 – Introduction to Compiler Construction: This has always been an implementation class in our department. The class implements new features in a scaled down, but running, mini-compiler. This differs from the build everything from scratch approach and is preferred by UW faculty because it models the typical industry situation of working with someone else's code. Students must “get their brain around” the compiler's structure before writing their own code. Further, it shows them good practice, as the mini-compiler is always beautifully written. Students extend the language by adding statements and operations, and extend the compilation by improving code generation, etc. They touch all phases from lexing to target code generation. They use source-control tools, lex-generator, parse-generator, and development tools like debuggers. Of course, along with all of the implementation, concepts, theory and best practices are covered in lecture. Past graduates have commented years after leaving campus that it was one of perhaps two most important classes for preparing them for a development career.

CSE403 – Software Engineering: Students complete a 9-week design and implementation project. The students propose ideas for software systems to build and are placed into groups of 5-8. Several deliverables are required during the course of the projects, including a requirements analysis, design specification, beta and final code releases, and testing materials. The design specification includes the submission of several documents and diagrams describing the system to be built. The students are taught and asked to use modern tools such as content management systems, design diagrams, and testing frameworks to generate their materials.

Examples of engineering design decisions students were required to make included:

- What features should the project have? What is the relative importance of these features, i.e., which ones should be cut if time becomes tight?
- What languages and tools should we use to design and implement these features? This includes CASE tools such as design diagram editors, schedulers, version control, bug tracking tools, as well as the overall programming language(s), APIs / libraries, and IDEs used to write the code.
- What should the testing plan be? This includes unit testing as well as system testing in some form.

CSE451 – Operating Systems: The objectives of the project component of CSE 451 are to a) provide experience in working in teams to design, implement, and evaluate important components of modern operating systems, b) provide experience in recognizing, evaluating,

and making the engineering design tradeoffs that characterize operating system design and, in fact, many other aspects of computer system design, c) provide experience in conducting performance experiments and understanding the performance tradeoffs that are crucial in operating system design and, again, in many other aspects of computer system design, and d) provide experience in producing coherent written technical descriptions and assessments of designs, design tradeoffs, and implementations.

The exact nature of the course depends on the operating system chosen as the exemplar. In most cases it is Linux, but in some offerings it is Windows. In both cases, students design features that have to be integrated with the existing, actual OS code base. In the most common instantiation of the course, students design and implement sophisticated components of modern systems, such as threads and synchronization primitives, resource managers, and components of the virtual memory and file systems.

Feedback from our alumni and their employers consistently indicates that this course is one of the most beneficial in our curriculum in terms of preparing students for engineering careers where they will tackle these sorts of issues on a regular basis.

CSE 461 – Introduction to Computer-Communications Networks: Students build substantial, layered protocols. Because the student projects must all interoperate, they need to all adhere to a protocol specification. In some cases, that specification is designed communally, though negotiation among all the teams. That is a very significant design experience, as it involves not only design and review but advocacy for one's position and convincing technical arguments for its superiority relative to other proposals. In other cases the protocol is specified by the course staff. In those cases, a reasonably deeply layered protocol stack must be built, layer by layer, and students are in charge of designing their implementations. The stakes are high, because each layer must support the ones to be built above it. In this version of the course, a basic mechanism is provided that allows students to design and implement extensions to the standard, primarily in an attempt to improve performance.

CSE466 – Embedded Systems: The design project for this class is built around programming a blimp, whose hardware was designed at UW. The blimp electronics includes multiple microcontrollers, a wireless link, sensors, and motor drivers. Students work in teams of 4 to complete an open ended design project with the blimp. Students typically implement a "remote control" mode where a PC (possibly connected to a game controller) wirelessly sends commands to the blimp. Students also usually implement an autonomous mode in which the blimp controls its height, heading, speed, or other parameters. The students design and implement their own radio protocol; their challenge is to maximize reliability, latency, and throughput. They also implement the wired communication protocol between the radio module on the blimp and the blimp controller. They implement (1) any PC-side software for communicating with or controlling the blimp, (2) the low level on-blimp code for reading sensor values and commanding motors and (3) the on-blimp autonomous control code. Students are graded both on technical challenge of their proposed functionality, as well as quality of execution, quality of documentation, and quality of presentation. Students are required to produce a web page with video documenting their blimp project, and to provide a live demo.

CSE467 – Advanced Digital Design: Students develop a full implementation of a complete hardware system that implements some real-time function like a graphics board or a video-rate image processor. In the image processing project, students started out by developing a video pipeline one component at a time. This comprised an interface to the video codec, followed by components that performed timing decoding, color space conversion and de-interlacing. Once the complete pipeline was complete, students then had the opportunity to implement a variety of real-time image transforms including scaling and rotation with bi-linear interpolation. This requires students to manage the complexity of a fairly large system using concepts of hierarchy, modularity and interface design. Students also gain substantial experience with state-of-the-art design tools for specification, simulation and synthesis, as well as modern FPGA architectures and hardware platforms. Our affiliates tell us that our students coming from this hardware design experience, complemented by the capstone course, have both the background and skills to become productive engineers almost immediately.

Required Capstone Design Courses

Our curriculum gives students an escalating role in design as they progress through it. This culminates with a major design experience in a capstone course. At least one design capstone is required for graduation. Since we began offering capstones in 1997, 89% of our computer engineering graduates have taken a single capstone, 9% have taken two, and 2% have taken three.

In capstone courses, students solve a substantial problem using concepts that span several topic areas in computer science and/or computer engineering. Students must work together in teams to define a problem, develop a solution plan, produce and demonstrate an artifact that solves the problem, and present their work. Class time focuses primarily on the project design and implementation, but it may also include lectures on the practical application of advanced topics. Cross-disciplinary projects that require interaction with other departments are encouraged.

Capstones are offered on a variety of topics, with new ones introduced regularly. During the last academic year, we had capstones on: internet systems; games; sound; resource-constrained environments; 3-d cameras; hardware systems; human-computer interface; and animation. We also offer capstones on computation biology; operating systems; home networking; robotics; and accessibility.

While capstones differ in topic, they adhere to a common set of themes and methods:

- **Integrate** - a team design experience makes concrete the foundational concepts introduced in related courses.
- Build on **student-accumulated knowledge and experience** to permit greater choice in the formulation of individual projects, even if they remain constrained in theme (e.g., a game, a race car, a networked appliance) or choice of implementation approach (software-only, embedded system, etc.).
- Introduce **human-centric design experiences** through projects that involve human-directed inputs based on system communicated outputs.

- Encourage the **development of communication skills** through frequent design presentations, project write-ups, and engagement within the project team and with external “customers.”
- Introduce **constrained design** by imposing user requirements and limitations to the implementation approach on the projects. Evaluate projects not simply on functionality (e.g., meeting functional requirements), but also on performance, cost/performance, and elegance (e.g., fewest components, lines of code, etc.).
- Use **cooperation** (e.g., sharing tools and libraries) and **competition** (e.g., bragging rights) where appropriate to undertake larger design or to motivate students.
- Re-conceptualize curriculum around the **project focus** to introduce materials “just in time” to be useful in project design and implementation.

We make videos that give some insight into each of the capstone courses and the types of projects that are developed. Those videos can be found at <http://www.cs.washington.edu/news/video/>.

As an example of the scope, organization, and activities in our capstones we show here the course page for the accessibility capstone from Winter 2011. (We chose this capstone simply because it has the pertinent information on a single web page, making it easy to include here. The information for other capstones is similar. We have deleted some material from the original page, <http://www.cs.washington.edu/education/courses/cse481h/11wi/>, for conciseness.)

Example: Accessibility Capstone Description, Winter Quarter 2011

As cell phones become more capable with connectivity with the internet and sensors such as cameras, compasses, GPS, and accelerometers, there are opportunities to use them as accessibility or assistive devices. In this capstone, students will work in teams to create new applications on cell phones that allow persons with disabilities to accomplish tasks that would be difficult to impossible to do without their applications. An example would be an application for a blind person that would take a picture of a bar code on a product, decode it, look it up on the internet, then speak the name of the product. Teams will then implement and test their concepts as working applications. Teams will prepare written reports on their applications, create videos, and present their applications in a public poster session where persons with disabilities will be invited.

Instructors and Mentors:

- Instructor: [Richard E. Ladner](#) (ladner at cs.washington.edu)
Office: CSE 632, 543-9347
- TA: [Shiri Azenkot](#) (shiri at cs.washington.edu)

Fall Preparation Seminar

11/12/10 4-5 pm [Introduction to Accessibility Capstone](#)
11/19/10 4-5 pm [Introduction 1 to Android Development](#)
12/3/10 4-5 pm [Introduction 2 to Android Development](#)
12/10/10 4-5 pm [Introduction 3 to Android Development](#)

Meeting Time:

- Tuesday and Thursday, 10:30 - 11:50, CSE 403

Tentative Schedule

- 01/04/11 - Project Assignments and Teams Assigned
- 01/06/11 - Project Reviews
- 01/11/11 - Concept Presentations - [Concept Presentation Form](#)
- 01/13/11 - Concept presentations to mentors
- 01/18/11 - Work plan presentations for feedback - [First Work Plan](#)
- 01/20/11 - Work plan presentations for feedback
- 01/25/11 - Team Meetings
- 01/27/11 - Team Meetings
- 02/01/11 - Team Meetings
- 02/03/11 - Team Meetings
- 02/08/11 - Team Meetings
- 02/10/11 - Team Meetings
- 02/15/11 - Prototype presentations with user feedback (2 hours)
- 02/17/11 - Team Meetings

- 02/22/11 - Team Meetings
- 02/24/11 - Team Meetings
- 03/01/11 - Team Meetings
- 03/03/11 - Poster Reviews
- 03/08/11 - Final Project Demonstrations
- 03/10/11 - Team Meetings without instructors
- 03/14/11 - Poster Session and Demonstrations (Monday 10:30 to 12:20 in the CSE Atrium)

Teams and Projects

- Team Gryffindor - Chris Raastad, Levi Lindsey, Michael Lam. Mentor: Shiri Azenkot. The Phone Wand - Navigating routes using orientation and vibration feedback.
- Kittens - Alison Wilbur, Katherine Tung, Junebae Kye. Mentor: Maria Kelley. Pic2Speech - Custom augmented speech.
- Mandroids - Jon Luo, Shurui Sun, Gary Kuo. Mentor: Chandrika Jayant. Street Sign Reader - Reading street signs so you don't have to.
- SEA - Euimin Jung, Siwei Kang, Angie Zhu. Mentor: Jessica Pitzer. WhosHere - Finding out which friends are around you.
- Team Synesthesia - Jon Ricaurte, Joy Kim. Mentor: Shaun Kane. TapBeats - The sound of gestures.
- Segfaults - Patrick Sweeney-Easter, Bradley Medlock. Mentor: Kristi Winter. Sound Detector - Detecting recognizable sound patterns to alert the user.

Posters

Posters should be 40" wide and 32" tall, as these will be easier to read. Do not use a font size smaller than 28 points. The amount of text should be minimal, with pictures (including captions that can be read who are blind or low-vision). Please use these [templates](#). Your poster should have a title, list of team members, and at least these sections: (i) Problem description, (ii) use example, (iii) description of solution, (iv) future work. You should include acknowledgements to your mentor and others who helped you. The poster should contain the UW and CSE logos and a mention that the poster is for the "Accessibility Capstone Design Course, Winter Quarter, 2010." There is a poster review on March 3rd.

Papers

Papers should use the two column format Microsoft Word [format](#) suggested by the ACM. The paper should be at most 4 pages and include the following sections: (i) Abstract which is a very short description of the project, like a tag line, (ii) introduction which includes target group, purpose of the project, and use case, (iii) related work which includes books, papers, products related to your project, (iv) description of the solution (subsections may be needed), (v) future work, (vi) references. Your paper may need additional sections. Papers are due by 10 pm on March 18th.

Videos

Videos will be approximately 3-4 minutes, describing the use of the accessibility application. Videos are due by 10 pm on Friday, March 18th.

Resources

- Related Projects
 - [MobileAccessibility Project](#) at the University of Washington
 - [AT&T speech mashup signup/login page](#)
 - [Project Possibility](#)
 - [IDEAL Group's Android Accessibility Project](#)
 - [WEKA open source machine learning engines](#)
 - [Wekinator open source machine learning related to music](#)
- Books for Android Developers
 - Hello, Android, Ed Burnette, ISBN# 978-1-934356-17-3 (Lots of good code samples; uses Sudoku game as example)
 - Beginning Android, Mark L. Murphy, ISBN# 978-1-4302-2419-8 (short chapters, each over a different aspect of Android)
 - Android Wireless Application Development (Developer's Library), Shane Conder & Lauren Darcey, ISBN# 978-0-321-62709-4 (more advanced topics)
- [UW Transit Lab](#)
- Lectures from the Accessibility Capstone Preparation Seminar, Autumn Quarter 2010
 - 9/30/09. Richard E. Ladner, CSE. [Persons with Disabilities](#)
 - 10/07/09. Richard E. Ladner, CSE. [Accessibility Technology and Research](#)
 - 10/14/09. Richard Mander, Entrepreneur in Residence University of Washington. [Assistive Technology Industry](#)
 - 10/21/09. Maria Kelley, UW Center for Technology and Disability Studies, Washington Assistive Technology Act Program. [Assistive Technology Services and Devices](#)
 - 10/28/09. Alan Borning, CSE. [OneBusAway](#)
 - 11/04/09. Shaun Kane, ISchool. [Survey on mobile devices used by persons with disabilities](#)
 - 11/18/09. Bruce Visser, Lighthouse for the Blind. [Technology for deaf-blind people](#)
 - 11/25/09. Shani Jayant, CSE. [MobileAccessibility Project](#)
 - 12/02/09. Anindya "Bapin" Bhattacharyya, Helen Keller National Center for Deaf-Blind Youths and Adults. Mobile technology for deaf-blind people.
 - 12/09/09. Shiri Azenkot, CSE. iPhone Accessibility Applications.
- Publications
 - [ACM Transactions on Accessible Computing \(TACCESS\) \(UW Access to ACM Digital Library\)](#)
 - [General Writing Guidelines for Technology and People with Disabilities](#) by Anna Cavender, Shari Trewin, Vicki Hanson
- Conferences
 - [Proceedings of the ACM SIGACCESS Conference on Assistive Technologies \(ASSETS\) \(UW Access to ACM Digital Library\)](#)
 - [Computers Helping People with Special Needs \(ICCHP\)\(UW Access\)](#)
 - [Assistive Technology Industry Association Conference](#)
 - [International Technology & Persons with Disabilities Conference \(CSUN\)](#)
- Organizations

- [World Federation of the Deaf](#)
- [National Association of the Deaf](#)
- [Alexander Graham Bell Association for the Deaf and Hard of Hearing](#)
- [National Federation of the Blind](#)
- [American Council of the Blind](#)
- [American Association of the Deaf-Blind](#)
- [Closing the Gap](#)
- [TRACE Center](#)
- [Smith-Kettlewell Eye Research Institute](#)
- News stories and web sites of interest to mobile accessibility
 - [Listening to Braille](#), New York Times, December 30, 2009.
 - [Insurers Fight Speech-Impairment Remedy](#), New York Times, September 14, 2009
 - ["Camera for the Blind"](#)
 - [WebAIM \(Web Accessibility in Mind\) article on low-vision](#)

Grading Criteria

- Functionality of the accessibility application(s) - Does it actually work as intended
- Quality of the code - Can the code be adopted by others as part of an open source effort
- Innovation - Is the application novel
- Impact - Does the application have impact on the lives of people with disabilities
- Quality of written report
- Quality of the poster and presentation
- Quality of the video
- Effort - Was the student's effort proportional to the overall team effort (A team is expected to have equal effort from all members)

Recent Capstone Offerings

We describe here some of the capstone projects that have been done in the past few years. These projects are often presented to the larger department community, including colleagues and industrial affiliates, in the form of live demonstrations, videotaped productions, and documentation published on the Web. The project reports and artifacts for representative design projects will be available for the ABET evaluation team.

CSE 428 – Computational Biology Capstone: In this capstone, students are taught the basic tools of computational biology, including molecular biology, biological sequence analysis, current computational tools and databases for computational molecular biology. They then apply their computer software engineering skills in teams to design, implement, and test a software system to perform high throughput analysis of a problem in molecular biology. In most cases, the project teams each contribute a part of the overall solution, and must co-ordinate their efforts to produce the final product. Example projects have included developing software that identifies evolutionarily conserved motifs in the DNA regulatory regions of homologous genes from multiple bacteria, phylogenetic footprinting in yeast species, and cataloging the prokaryotic regulatory elements.

CSE 454 Advanced Internet Systems Capstone: Students combine techniques in basic Internet programming with concepts from information retrieval, machine learning and data-mining to construct scalable and secure Web-based services for searching and classifying Web-based content. In one example project, students constructed a large-scale Google-style search engine with a focus on using data-mining techniques to provide high-quality ranking for large numbers of search results. Other examples include a specialized search engine for webcams, event-detection from the Twitter stream, and comprehensive information extraction systems.

CSE 477 Digital System Design Capstone: This course has a strong focus on hardware/software integration and embedded systems design. This includes hardware-intensive projects that are implemented using large FPGA platforms or distributed embedded sensor platforms. Students apply significant amount of knowledge gained from their embedded systems prerequisites and low-level programming language classes. In addition, projects must comprise of a combination of hardware and software components and leverage the different expertise of the project team members. Thus, students apply general software engineering practices, database and networking knowledge, interface design, and other specialized computer science concepts to build end-to-end final projects.

CSE 481C – Robotics Capstone: This capstone covers concepts in Robotics, culminating in a capstone project that applies these concepts to a sizeable problem. These concepts include behavior-based techniques, probabilistic techniques, particle filters, Markov design processes and active sensing. Each offering poses an open-ended problem that students work on in groups. The objective of the most recent offering was programming a NAO humanoid robot to imitate human actions and learn new skills from human demonstration with the help of a Kinect RGB+depth camera. Three teams of four students each focused on various aspects of (1) human motion capture and interpretation from video, (2) control of the humanoid robot, and (3) implementation of probabilistic reasoning and machine learning algorithms for learning from human demonstration.

CSE 481D Computer Games Capstone: This course is centered on giving students experience with hands-on design and development of a substantial software product. In the capstone project, students learn how to use modern, real-world tools to build complex software systems. The games domain requires students to deal with the complexity and performance demands of high-performance, real-time, distributed application. Student teams implement games of their own design, so that they are involved with the creative process from start to finish. At the end of the course, students present and demonstrate their work to a larger department audience.

CSE 481H – Accessibility Capstone: As smartphones become more capable with connectivity with the internet and sensors such as cameras, microphones, compasses, GPS, gyroscopes, and accelerometers, there are opportunities to use them as accessibility or assistive devices. In this capstone, students will work in teams to create new applications on smartphones that allow persons with disabilities to accomplish tasks that would be difficult to impossible to do without their applications. An example would be an application for a blind person that would take a picture of a bar code on a product, decode it, look it up on the internet, then speak the name of the product. Teams will then implement and test their concepts as working applications with potential users. Students are expected to apply knowledge they have learned in classes such as network programming, computer vision, signal processing, and human computer interaction to create their solutions. Teams will prepare written reports on their applications, create videos, and present their applications in a public poster session where persons with disabilities will be invited.

CSE481I – Sound Capstone: This capstone focuses on building digital musical instrument prototypes utilizing computer audio techniques for sound synthesis, recognition, and analysis/re-synthesis, coupled with human gestural capture and embedded system hardware design. Students work in teams to design, implement, and release a hardware/software project utilizing some of the techniques surveyed. Example projects have included a Musical Glove interface, a Kinect-based virtual drum kit, and a Cochlear-implant simulator.

CSE481K – Designing Technology for Resource-Constrained Environments Capstone: In CSE 481K, students build hardware or software solutions to problems faced in local or remote resource-constrained environments (e.g. the developing world or low-income settings). These environments provide unique constraints (e.g., cultures where people are unfamiliar with or afraid of technology, environments where power and network connectivity are scarce and/or expensive). Students work in groups to create testable prototypes and complete initial evaluations of the software and hardware artifacts produced. Projects vary from year to year but are based on real-world problems with real customers who provide feedback to the development team. Examples of solutions built in the past include: smart phone application for digitizing paper forms, SMS-based vehicle tracking system for public transportation, and a low cost interface for antenatal ultrasound. These three examples and several others have led to publications in workshops and conferences as well as to actual deployments - although these are beyond the scope of the course and usually happen as the students continue to further refine their projects after the course is over.

Progress is assessed via weekly meetings and groups document their work in the form of posters, web pages, verbal presentations, videos, and written reports.

The course is paired with a 2 credit design studio held in the preceding quarter. While this course is optional, the majority of students who take the capstone also complete the design

studio. It is a free elective course. In the design studio, students form interdisciplinary groups to scope and design projects. Several of the projects scoped in the design studio continue on to be implemented and evaluated in the capstone course. Although not strictly required, typically students sign up for both the design studio held in the preceding quarter and the capstone course. Recent offerings of the course have involved close work with students and faculty from the Department of Human-Centered Design & Engineering

A.7 Extra-Curricular Activities and Co-op

A.7.1 Extra-Curricular Activities

We also use extra-curricular elements to further accentuate outcomes that are particularly important to engineers practicing in our field or continuing on to graduate school. These are:

- Industry experience by means of co-ops and internships
- Research participation
- Teaching assistantships

Although our students are not required to participate in these extra-curricular elements, a large number of them do, and gain the most from what our type of department has to offer. For example, more than 80% of our students are recorded as having completed industry internships. Nearly 50% of our students participate in research with faculty and graduate students. Many of our undergraduates also publish research papers with faculty.

Finally, approximately 35% of our graduating class each year has participated as teaching assistants or teaching consultants not only for our introductory courses but for many of our capstone and other laboratory intensive courses as well. We find that our own undergraduates, who are familiar with our equipment and development environments, are often more effective than graduate students, who have not worked directly with the same tools.

We strive to increase participation in these co-curricular activities via tech talks and research nights run by the ACM student chapters, career and internship preparation events, orientation comments and explanations by the Undergraduate Program Coordinator and Chair, through our Undergraduate Advising Unit, and through student mentoring. The participation rate has been steadily rising as students become aware of these options and talk to fellow students who have benefited from their experiences. Employers increasingly view student participation in research and teaching assistantships as a positive when reviewing applicants for employment.

A.7.2 Co-op Credits

Our co-op requirements and credits are described in Section 1E.

A.8 Materials Available to Committee

The vast majority of course materials are managed online; assignments are posted online; course notes are posted online; solutions are posted online; student work submission is online. We expect to make the majority of these items available through local web access. (Access is local because, especially in the case of work samples, there are security concerns related to confidentiality.)

In addition, we will have copies of textbooks used in our courses.

B. Course Syllabi

Course syllabi are in Appendix A.

CRITERION 6. FACULTY

A. *Faculty Qualifications*

The department operates with minimal hierarchy. -Faculty of all ranks have similar teaching loads. All faculty have annual peer reviews of teaching and meetings with the Chair to discuss activity level and performance in teaching, research, and service. Annual evaluations may include a reflective statement from each faculty member to the Chair outlining their assessment of the past year and goals for the future.

There are 49 tenured or tenure track faculty:

- 24 full professors (1 joint appt)
- 16 associate professors (2 joint appts)
- 9 assistant professors (3 joint appts)

Lecturers

- 1 principal lecturer
- 3 senior lecturers
- 2 lecturers

The CSE faculty covers a wide range of the discipline and, collectively, has a wide range of experience in our discipline including: industry research labs, start-ups, and advanced development. Faculty members are highly visible in their respective communities with many serving as chairs of major conferences and enjoying extensive networks of colleagues on an international, as well as national, level.

Thirty-nine current and past CSE faculty members have won Presidential/NSF Young Investigator Awards or NSF CAREER Awards. Two have won MacArthur Fellowships. Six faculty members are ONR Young Investigator Award recipients. Six hold NSF Presidential Faculty Fellow or Presidential Early Career (PECASE) Awards. One holds the NSF Presidential Excellence in Science, Mathematics and Engineering Mentoring (PAESMEM) Award. Twenty-one (plus three Adjunct faculty) hold Sloan Research Fellowships. Three hold Packard Fellowships and three hold Microsoft Research Faculty Fellowships. Among the senior faculty are eight Fulbright recipients, two Guggenheim recipients, eighteen Fellows of the ACM, thirteen Fellows of the IEEE, two Fellows of the International Association for Pattern Recognition, five Fellows of the American Association for Artificial Intelligence, three Fellows of the American Association for the Advancement of Science, one Fellow of the American Academy of Arts & Sciences, and three (plus six Adjunct/Affiliate) Members of the National Academies.

All faculty are involved with teaching in both our undergraduate programs. Eight faculty members in the department are recipients of University awards: 5 are winners of the Distinguished Teaching Award (Borriello, Diorio, Ebeling, Salesin, and Reges), 1 is the winner of the Distinguished Graduate Mentor Award (Notkin), and 2 received Outstanding Public Service Awards (Ladner and Lazowska). Three faculty members are recipients of the College of

Engineering's Faculty Achievement Award (Borriello, Ebeling, and Salesin). In 1999, CSE received the inaugural UW Brotman Award for Instructional Excellence -- in essence a departmental distinguished teaching award. In 2007, Richard Anderson received the College of Engineering award for Innovation in Education. The ACM student chapter honors a faculty member at our departmental graduation event with an ACM Distinguished Teaching Award decided by a student vote. Finally, the Bob Bandes award honors the Department's top TAs based on nominations from students.

Curriculum quality is enhanced through faculty self-assessments of their teaching performance after each course, peer evaluations of teaching for faculty of all ranks on an annual basis, and student evaluations of each course. The self and peer evaluations touch upon not only teaching performance in specific course offerings but also consider curriculum development, development of course infrastructure, future plans for the course content and assignments, and how previously identified issues were handled.

Our faculty continually strives to improve its teaching performance. All our courses are evaluated by students. Peer evaluations of teaching are taken very seriously. Graduate student instructors are provided with a seminar to develop their skills (CSE 590IT). The faculty takes full advantage of the resources the University and College have to offer. In particular, many of our faculty have had representatives from the Center for Teaching and Learning (CTL – <http://www.washington.edu/teaching/>) visit their classrooms and conduct student interviews and make suggestions for improving their lectures. Interactions with the College of Engineering's Center for Engineering Learning and Teaching (CELT – depts.washington.edu/celtweb/) have included: individual faculty consultations with instructional consultants; faculty participation in CELT workshops and attendance at CELT sponsored talks; collaborations in educational research proposals; and evaluation of educational projects.

The faculty is extremely active in research and our department is consistently ranked in the top 10 in the nation in our discipline. There are strong ties with industry with many faculty receiving research and educational support from corporations, through consulting, and past experience in industry. National visibility is high with several faculty members serving on National Science Foundation advisory boards, National Research Council study panels, and Department of Defense research organizations. One of our emeritus faculty (Snyder) is the leader in a drive to develop university and high school courses in information technology literacy (see our own CSE 120).

Within the University, three faculty members have received the College of Engineering Faculty Achievement Award. In 1996, Ed Lazowska became the first member of the College of Engineering to be named the University of Washington Annual Faculty Lecturer, and in 1998 he received the University of Washington Outstanding Public Service Award.

Table 6-1 Faculty Qualifications

Computer Engineering faculty highlighted

Faculty Name	Highest Degree Earned- Field and Year		Type of Academic Appointment ² T, TT, NTT	FT or PT ³	Years of Experience			Professional Registration/ Certification	Level of Activity ⁴ H, M, or L		
					Govt./Ind. Practice	Teaching	This Institution		Professional Organizations	Professional Development	Consulting/summer work in industry
Anderson, Richard	PhD, Computer Science Stanford, 1985	P	T	FT		26	26		L	M	N
Anderson, Ruth	Ph. D., Computer Science University of Washington, 2006	O	NTT	FT							
Anderson, Tom	PhD, , Computer Science Washington, 1991	P	T	FT	4	21	15		M	M	L
Balazinska, Magdalena	PhD, , Computer Science MIT, 2006	ASC	T	FT		7	7		H	H	N
Beame, Paul	PhD, , Computer Science Toronto, 1987	P	T	FT		25	25		M	H	L
Borning, Alan	PhD, Computer Science Stanford 1979	P	T	FT	3	32	32		M	H	L
Borriello, Gaetano	PhD, EE & Computer Science UC Berkeley, 1988	P	T	FT	3	25	25		L	H	M
Ceze, Luis	PhD, Computer Science Uillinois Urbana-Champaign, 2007	ASC	T	FT		5	5			H	
Curless, Brian	PhD, Electrical Engineering Stanford, 1997	P	T	FT	0	14	14		L	H	L
Domingos, Pedro	PhD, Computer Science UC-Irvine, 1997	P	T	FT		15	13		L	H	L
Ernst, Michael	PhD, Computer Science Washington, 2000	ASC	T	FT		12	4			H	
Etzioni, Oren	PhD, Computer Science Carnegie Mellon 1990	P	T	FT		21	21		M	M	M

Faculty Name	Highest Degree Earned- Field and Year		Type of Academic Appointment ² T, TT, NTT	FT or PT ³	Years of Experience			Professional Registration/ Certification	Level of Activity ⁴ H, M, or L		
					Govt./Ind. Practice	Teaching	This Institution		Professional Organizations	Professional Development	Consulting/summer work in industry
Farhadi, Ali	PhD, Computer Science U Illinois Urbana-Champaign, 2011	AST	TT	FT						H	
Fogarty, James	PhD, Computer Science Carnegie Mellon, 2006	ASC	T	FT		6	6			H	
Fox, Dieter	PhD, Computer Science U Bonn, 1998	ASC	T	FT	2	14	12			H	
Gollakota, Shyamnath	PhD, EE & Computer Science, MIT, 2013	AST	TT	FT		0	0			H	
Gribble, Steve	PhD, Computer Science UC Berkeley, 2000	P	T	FT		12	12		L	H	N
Grossman, Daniel	PhD, Computer Science Cornell, 2003	ASC	T	FT		9	9		L	H	L
Guestrin, Carlos	PhD, Computer Science Stanford, 2003	ASC	T	FT	1	8	0			H	
Hemingway, Bruce	AB, Music Indiana U, 1973	O	NTT	PT	32	12	10		L	L	L
Karlin, Anna	PhD, Computer Science, Stanford, 1987	P	T	FT	6	18	18		M	H	N
Kemelmacher, Ira	PhD, Computer Science & Appl Math Weizmann Inst. Of Science, 2009	AST	TT	FT	1	0	0			H	
Kohno, Tadayoshi	PhD, Computer Science UC San Diego, 2006	ASC	T	FT		5	5			H	
Krishnamurthy, Arvind	PhD, Computer Science UC Berkeley, 1999	ASC	T	FT		13	7			H	
Ladner, Richard	PhD, Mathematics UC Berkeley, 1971	P	T	FT		41	41		L	H	M

Faculty Name	Highest Degree Earned- Field and Year		Type of Academic Appointment ² T, TT, NTT	FT or PT ³	Years of Experience			Professional Registration/ Certification	Level of Activity ⁴ H, M, or L		
					Govt./Ind. Practice	Teaching	This Institution		Professional Organizations	Professional Development	Consulting/summer work in industry
Landay, James	PhD, Computer Science Carnegie Mellon, 1996	P	T	FT	2	16	8		L	H	M
Lazowska, Ed	PhD, Computer Science U Toronto, 1977	P	T	FT	0	35	35		H	L	H
Lee, James R	PhD, Computer Science UC Berkeley, 2005	ASC	T	FT	0	6	6		L	H	N
Lee, Su-In	PhD, Computational Biology Stanford, 2010	AST	TT	Joint FT		4	3			H	
Levy, Henry M	MS, Computer Science Washington, 1981	P	T	FT	8	29	29		H	L	H
Martin, Helene	BS, Computer Science, Washington, 2008	O	NTT	FT		3			L	H	L
Mausam	PhD, Computer Science & Eng Washington, 2007	O	NTT	FT	0	0	5			H	
Mones, Barbara	MFA, Sculpture Rhode Island Schl of Design, 1979	O	NTT	FT	3	26	13		H	L	M
Oskin, Mark	PhD, Computer Science UC Davis, 2001	ASC	T	FT	2	12	12		H	H	M
Patel, Shwetak	PhD, Computer Science Georgia Tech, 2008	AST	TT	Joint FT	1	4	4			H	H
Perkins, Hal	MS, Computer Science Cornell, 1982	O	NTT	FT	2	31	14		L	N	N
Popovic, Zoran	PhD, Computer Science Carnegie Mellon, 1999	P	T	FT	0	13	13		N	H	N
Rao, Anup	PhD, Computer Science U Texas, Austin, 2007	AST	TT	FT		3	3			H	
Rao, Rajesh	PhD, Computer Science U Rochester, 1998	ASC	T	FT	2	12	12		M	H	L

Faculty Name	Highest Degree Earned- Field and Year		Type of Academic Appointment ² T, TT, NTT	FT or PT ³	Years of Experience			Professional Registration/ Certification	Level of Activity ⁴ H, M, or L		
					Govt./Ind. Practice	Teaching	This Institution		Professional Organizations	Professional Development	Consulting/summer work in industry
Reges, Stuart	MS, Computer Science Stanford, 1982	O	NTT	FT	0	30	8		M	L	L
Ruzzo, Larry	PhD, Computer Science UC Berkeley, 1978	P	T	FT	0	35	35		N	H	N
Seelig, Georg	PhD, Physics U of Geneva, 2003	AST	TT	Joint FT			3			H	
Seitz, Steve	PhD, Computer Science U Wisconsin, 1997	P	T	FT	3	15	12		L	H	L
Shapiro, Linda	PhD, Computer Science U Iowa, 1974	P	T	FT Joint	0	38	26		M	H	L
Smith, Joshua	PhD, Media Arts & Sciences MIT, 1999	ASC	TT	FT- Joint	9	2	2			H	
Stepp, Martin	MS, Computer Science U Arizona, 2003	O	NTT	FT	1	9	6				
Suciu, Dan	PhD, Computer Science U Pennsylvania, 1995	P	T	FT	5	22	12		L	H	M
Tanimoto, Steve	PhD, Computer Science Princeton, 1975	P	T	FT	0	37	35		M	H	H
Taskar, Benjamin	PhD, Computer Science Stanford, 2005	ASC	T	FT		5	0			H	
Todorov, Emanuil	PhD, Cognitive Neuroscience MIT, 1998	ASC	T	FT		10	3			H	
Tompa, Martin	PhD, Computer Science U Toronto, 1978	P	T	FT		34	34		L	H	H
Weld, Daniel	PhD, Artificial Intelligence MIT, 1988	P	T	FT		24	24		H	H	H
Wetherall, David	PhD, Computer Science MIT, 1998	P	T	FT	3	12	12		N	H	L

Faculty Name	Highest Degree Earned- Field and Year		Type of Academic Appointment ² T, TT, NTT	FT or PT ³	Years of Experience			Professional Registration/ Certification	Level of Activity ⁴ H, M, or L		
					Govt./Ind. Practice	Teaching	This Institution		Professional Organizations	Professional Development	Consulting/summer work in industry
Zahorjan, John	PhD, Computer Science U Toronto, 1980	P	T	FT		32	32		L	M	L
Zettlemoyer, Luke	PhD, Computer Science MIT, 2009	AST	TT	FT		3	3			H	

Instructions: Complete table for each member of the faculty in the program. Add additional rows or use additional sheets if necessary. Updated information is to be provided at the time of the visit.

1. Code: P = Professor ASC = Associate Professor AST = Assistant Professor I = Instructor A = Adjunct O = Other
2. Code: T = Tenured TT = Tenure Track NTT = Non Tenure Track
3. Code: FT = Full-time PT = Part-time Appointment at the institution.
4. The level of activity (high, medium or low) should reflect an average over the year prior to the visit plus the two previous years.

B. Faculty Workload

Table 6-2 Faculty Workload Summary

Faculty Member (name)	PT or FT ¹	Classes Taught (Course No./Credit Hrs.) Term and Year ²	Program Activity Distribution ³			% of Time Devoted to the Program ⁵
			Teaching	Research or Scholarship	Other ⁴	
Anderson, Richard	FT	12au CSE 311 (4)	11%	72%	17%	100%
Anderson, Ruth	FT	12au CSE 390A (1), CSE 373 (3) 13wi CSE 390A (1), CSE 490D (2) CSE 332 (4) 13sp CSE 390A(1), CSE 332 (4)	70%	20%	10%	100%
Anderson, Tom	FT	12au CSE 451 (4) 13sp CSE 452 (4)	22%	61%	17%	100%
Balazinska, Magdalena	FT	12au CSE 344 (4) 13sp CSE 444 (4)	22%	61%	17%	100%
Beame, Paul	FT	12au CSE 421 (3) 13sp CSE 311 (4)	22%	61%	17%	100%
Borning, Alan	FT	12au CSE 341 (4) 13wi CSE 510 (4)	22%	61%	17%	100%
Borriello, Gaetano	FT	12au CSE 351 (4)	11%	72%	17%	100%
Ceze, Luis	FT	12au CSE 548 (4) 13sp CSE 351 (4)	22%	61%	17%	100%
Curless, Brian	FT	13sp CSE 457 (4)	11%	72%	17%	100%
Domingos, Pedro	FT				100%	100%
Ernst, Michael	FT	13wi CSE 140 (4) 13sp CSE 331 (4)	22%	61%	17%	100%

Faculty Member (name)	PT or FT ¹	Classes Taught (Course No./Credit Hrs.) Term and Year ²	Program Activity Distribution ³			% of Time Devoted to the Program ⁵
			Teaching	Research or Scholarship	Other ⁴	
Etzioni, Oren	FT				100%	100%
Farhadi, Ali	FT				100%	100%
Fogarty, James	FT				100%	100%
Fox, Dieter	FT	12au CSE 481 (5)	11%	72%	17%	100%
Gollakota, Shyamnath	FT	13sp CSE 461 (4)	11%	72%	17%	100%
Gribble, Steve	FT	12au CSE 333 (4)	11%	72%	17%	100%
Grossman, Daniel	FT	12au CSE 505 (4) 13wi CSE 341 (4) 13sp CSE 341 (4)	33%	50%	17%	100%
Guestrin, Carlos	FT	13sp CSE 446 (3)	11%	72%	17%	100%
Hemingway, Bruce	FT	12au CSE 131 (4) 13wi CSE 481 (5) 13sp CSE 131 (4)	33%	50%	17%	100%
Karlin, Anna	FT	13wi CSE 312 (4) 13sp CSE 525 (4)	22%	61%	17%	100%
Kemelmacher, Ira	FT	13sp CSE 576 (3)	11%	72%	17%	100%
Kohno, Tadayoshi	FT	13wi CSE 484 (4)	11%	72%	17%	100%
Krishnamurthy, Arvind	FT	12au CSE 550 (4) 13wi CSE 461 (4)	22%	61%	17%	100%
Ladner, Richard	FT			50%	50%	100%
Landay, James	FT	12au CSE 440 (5) 13wi CSE 440 (5) 13sp CSE 441 (5)	33%	50%	17%	100%

Faculty Member (name)	PT or FT ¹	Classes Taught (Course No./Credit Hrs.) Term and Year ²	Program Activity Distribution ³			% of Time Devoted to the Program ⁵
			Teaching	Research or Scholarship	Other ⁴	
Lazowska, Ed	FT	13sp CSE 451 (4)	11%	72%	17%	100%
Lee, James R	FT				100%	100%
Lee, Su-In	FT	12au CSE 527 (4)	11%	72%	17%	100%
Levy, Henry M	FT		5%	35%	60%	100%
Martin, Helene	FT	12au CSE 142 (4) 13wi CSE 143 (5)	22%		78%	100%
Mausam	FT			80%	20%	100%
Mones, Barbara	FT	12au CSE 458 (5) 12au CSE 459 (5) 13wi CSE 460 (5) 13sp CSE 464 (5)	44%	39%	17%	100%
Oskin, Mark	FT	12au CSE 352 (4) 13wi CSE 458 (4) 13sp CSE 352 (4)	33%	50%	17%	100%
Patel, Shwetak	FT	13sp CSE 477 (5)	11%	72%	17%	100%
Perkins, Hal	FT	12au CSE 331 (4) 12au CSE 413 (3) 13wi CSE 331 (4) 13wi CSE 374 (3) 13sp CSE 333 (4) 13sp CSE 414 (4)	90%		10%	100%
Popovic, Zoran	FT				100%	100%
Rao, Anup	FT	13wi CSE 421 (3) 13sp CSE 431 (3)	22%	61%	17%	100%

Faculty Member (name)	PT or FT ¹	Classes Taught (Course No./Credit Hrs.) Term and Year ²	Program Activity Distribution ³			% of Time Devoted to the Program ⁵
			Teaching	Research or Scholarship	Other ⁴	
Rao, Rajesh	FT	12au CSE 473 (3) 13sp CSE 528 (3)	22%	61%	17%	100%
Reges, Stuart	FT	12au CSE 143 (5) 13wi CSE 142 (4) 13sp CSE 143 (5)	60%		40%	100%
Ruzzo, Larry	FT	12au CSE 312 (4) 13wi CSE 521 (4)	22%	61%	17%	100%
Seelig, Georg	FT	13wi CSE 488 (4)	11%	72%	17%	100%
Seitz, Steve	FT	12au CSE 455 (4)	11%	72%	17%	100%
Shapiro, Linda	FT	13sp CSE 373 (3)	11%	72%	17%	
Smith, Joshua	FT	12au CSE 466 (4)	11%	72%	17%	100%
Stepp, Martin	FT	12au CSE 154 (5) 13wi CSE 373 (3) 13sp CSE 154 (5) 13sp CSE 403 (4)	44%		56%	100%
Suciu, Dan	FT	12au CSE 311 (4) 13wi CSE 344 (4) 13sp CSE 544 (4)	33%	50%	17%	100%
Tanimoto, Steve	FT	12au CSE 415 (3) 13wi CSE 417 (3)	22%	61%	17%	100%
Taskar, Benjamin	FT	13sp CSE 515 (4)	11%	72%	17%	100%
Todorov, Emanuil	FT	13wi CSE 529 (3)	11%	72%	17%	100%
Tompa, Martin	FT	12au CSE 332 (4) 13wi CSE 427 (3)	33%	50%	17%	100%

Faculty Member (name)	PT or FT ¹	Classes Taught (Course No./Credit Hrs.) Term and Year ²	Program Activity Distribution ³			% of Time Devoted to the Program ⁵
			Teaching	Research or Scholarship	Other ⁴	
		13sp CSE 312 (4)				
Weld, Daniel	FT	12au CSE 573 (4) 13wi CSE 454 (5)	22%	61%	17%	100%
Wetherall, David	FT				100%	100%
Zahorjan, John	FT	12au CSE 461 (4) 13wi CSE 333 (4) 13sp CSE 481 (5)	33%	50%	17%	100%
Zettlemoyer, Luke	FT	13wi CSE 517 (4) 13sp CSE 473 (3)	22%	61%	17%	100%

1. FT = Full Time Faculty or PT = Part Time Faculty, at the institution
2. For the academic year for which the self-study is being prepared.
3. Program activity distribution should be in percent of effort in the program and should total 100%.
4. Indicate sabbatical leave, etc., under "Other."
5. Out of the total time employed at the institution.

C. Faculty Size

The department has about 49 tenure-line faculty members, 6 instructors, and around 35 postdocs.

Since 2008 the department has made a number of significant hires that greatly strengthen our computer engineering program:

1. Michael Ernst (Associate Professor, formerly at MIT) is an expert in software engineering, programming languages, and computer security.
2. Emo Todorov (Associate Professor, formerly at UCSD) is known for his application of control theory to robotics and neuromuscular movement.
3. Shwetak Patel (Assistant Professor, recently promoted to Associate with tenure) develops new sensor technologies for energy monitoring, health care, ubiquitous computing, and user interfaces. Patel received a MacArthur Award in 2011. He is joint with Electrical Engineering in the ExCEL program (“Experimental Computer Engineering Laboratory”).
4. Georg Seelig (Assistant Professor) works in synthetic biology. He was first to demonstrate how to create simple logic gates out of DNA molecules. He is joint with Electrical Engineering in ExCEL.
5. Luke Zettlemoyer (Assistant Professor) works in the area of natural language processing and machine learning.
6. Josh Smith (Associate Professor) was hired from Intel Research in Seattle. Smith specializes in developing novel sensors and wireless power systems (systems that compute and communicate without any onboard power source). He is the inventor of WISP, a Wireless Identification and Sensing Platform based on RFID technology. He has a joint appointment in Electrical Engineering as part of ExCEL.
7. Arvind Krishnamurthy (Associate Professor) is an expert in distributed systems and computer networks.
8. Shyam Gollakota (Assistant Professor) specializes in networks and novel uses of wireless networks. He recently won the 2012 ACM Doctoral Dissertation Award for his MIT thesis on “Embracing interference in wireless systems.”
9. Ali Farhadi (Assistant Professor) works in computer vision, particularly new developments in object recognition.
10. Ira Kemelmacher (Assistant Professor) works in computer vision, specializing in facial processing and modeling. Her technology is the basis of Google/Picassa Face Movie.

In addition, we currently have an offer outstanding to a senior candidate in sensor technology and wireless systems, jointly with Electrical Engineering, that is still in process.

D. Professional Development

In CSE, sabbaticals are permitted every 7 years and are supported by providing 2/3s salary over three quarters. The department has very liberal policies with regard to faculty leaves to spend time in industry or to spinout department-developed technology (e.g., in startups). In general, we support faculty members' requests for industrial interaction because we believe it enhances both our educational and research missions, creates new connections to industry, and gives faculty insight into important current problems faced by industry. Our Industrial Affiliates Program provides connections with over 100 high-tech companies both locally and nationally.

CSE provides support for staff members, instructors, and students to travel to workshops, seminars, and meetings. We also support the creation of campus-wide technical communities in areas of importance to us.

E. Authority and Responsibility of Faculty

The department leadership consists of the Chair (Hank Levy), two Associate Chairs (Paul Beame, who focuses on educational activities, and Ed Lazowska, who focuses on development and external relations), and an Executive Committee consisting of these three individuals plus Alan Borning, Gaetano Borriello, Dan Grossman, James Lee, and Georg Seelig.

The Chair is appointed by the Dean for a five-year term; Levy was appointed in 2006 and recently agreed to a two-year extension. To provide some stability while the College found and installed a new Dean, he has agreed to an additional year. The Associate Chairs are appointed by the Chair for short terms (one or more years) and for duties that correspond with current departmental needs and individual interests. Additional members of the Executive Committee are elected for one year terms with a one-year reprieve after two consecutive terms.

The general duties of the Executive Committee are to be "in the loop" on all issues, to deal with straightforward issues without engaging the department as a whole, to approve simple appointments (e.g., Affiliate Faculty and postdocs), to ensure that the department as a whole is engaged on important issues, and to serve as a two-way communication channel.

CRITERION 7. FACILITIES

A. Offices, Classrooms and Laboratories

Offices & Classrooms

The department moved into the Paul Allen Center in 2003, greatly expanding space for labs and meeting rooms. The Paul Allen Center is adjacent to the recently-completed building for the Electrical Engineering Department and shares a large atrium that serves as a "student commons". There is a total of 85,000 square feet of space – this new building increased CSE's total space by a factor of 2.5 and its laboratory space by a factor of 3, for the first time approaching national norms for comparable programs.

CSE has 14,000 square feet of faculty offices. These include teaching faculty, research, tenure and non-tenure emeriti. 11,000 square feet of office and office type space is used for staff-business office, advisors, technical support, research staff, grant coordinators, general department support. 13,000 square feet of office space dedicated to our PhD candidates involved in research and teaching.

While there are no classrooms in the Paul G Allen for CSE, the department shares lab space with the I-School in a separate building- Introductory Programming Learning Commons, 1,500 square feet with 47 total work stations. The department also maintains roughly 8,000 square feet of space on the 3rd floor of our previous home, Sieg Hall.

Computer Labs

A total of 10,062 square feet is devoted to CSE lab space for general computing, animation courses, hardware design and special projects.

Instructional computing is provided through laboratories operated within the department. These include three general use laboratories with over 75 Intel PCs running Windows 7 and Fedora Core Linux. In order to facilitate student mobile computing and personal devices, two of these labs are equipped "power-up" stations that can accommodate power and physical Ethernet connection for an additional 50 personal devices (laptops, etc). Students also have access to a dual-band wireless network that provides 802.11b/g/n connectivity throughout the building and in surrounding exterior areas. Additional back-end resources are provided by Intel PC-based compute and file servers running Linux. Other remote instructional computing facilities include the availability of a downloadable Linux virtual machine that includes a development environment very similar to what is provided in the general computing laboratories. The department also maintains a remotely accessible Windows "Virtual Lab" built on a customized

VDI platform with 18 concurrent connections, or “seats”, available to students. These hosted VMs also contain a nearly identical software image to what is offered in our laboratories.

The department operates four special-purpose laboratories containing approximately 100 Intel workstations. To support digital system design courses, the Embedded Systems Laboratory contains 54 Intel-based workstations for design entry and simulation along with Tektronix logic analyzers, digital oscilloscopes and other test equipment. Undergraduate capstone courses utilize the following lab facilities. The Laboratory for Animation Arts includes 24 Intel workstations and digital video production equipment, and is used for teaching interdisciplinary courses in computer animation. The Special Projects Lab and Capstone Computing Lab have capacity for 20 and 12 workstations respectively, and are customizable in their physical setup. They are frequently used to teach a variety of CSE capstone courses including; digital game design, home networking and robotics, which often require specialized equipment, unique software, and/or dedicated access.

These lab spaces are available 24 hours a day, 7 days a week, to all CSE students and others enrolled in CSE major and graduate courses, subject to weekly maintenance, course restriction, or existing reservations. The weekly maintenance and lab closure schedule can be viewed on the CSE Instructional Lab site: <http://www.cs.washington.edu/lab/facilities/instr-labs/>.

General Computing Labs (4400 sq. ft.)

- Workstation Lab (CSE 002) – 13 Windows PCs, 15 Linux PCs, desks can accommodate physical connection of 38 personal laptops/devices
- Workstation Lab (CSE 006) – 3 Windows PCs, 20 Linux PCs, desks can accommodate physical connection of 12 personal laptops/devices
- Workstation Lab (CSE 022) – 20 Windows PCs, 6 Linux PCs

Specialized Computing Labs (2189 sq. ft.)

- Digital Animation Lab (Sieg 329) – 24 workstations, 28-node RenderFarm (capable of bursting up to 50 nodes utilizing commodity Cloud vendor services)
- Capstone Computing Lab (CSE 003D) – 12 workstations (at full capacity)
- Special Projects Lab (Sieg 327) – 20 workstations (at full capacity)

Hardware Design Labs (3473 sq. ft.)

- Hardware Lab (CSE 003) – 39 Workstations

- Hardware Projects Lab (CSE 003E) – 13 Workstations

The hardware labs (CSE 003/003E) additionally contain many test instruments and hardware platforms for design of hardware projects and embedded systems. The main hardware lab (CSE 003) has 14 workbenches, each equipped with a workstation, oscilloscope, logic analyzer, function generator and power supply. The hardware projects lab (CSE 003E) has an additional 12 workbenches, 11 of which are equipped with a workstation, oscilloscope, logic analyzer, function generator and power supply. Additional test equipment like multimeters and logic programmers are also available.

A variety of hardware design platforms are used in classes. The introductory digital design course uses a breadboard attached to lights and switches, as well as an Altera DE1 platform FPGA that is used for clock generation and testing. The advanced courses use a variety of FPGA-based platforms, particularly the Altera DE1 platform that contains a Cyclone II 20 million gate FPGA as well as a variety of interfaces.

Complete information about the hardware lab facilities is available online at:

<http://www.cs.washington.edu/lab/facilities/hwlab/>

Design Tools for Hardware Design

A large variety of software packages are used by the hardware oriented courses:

- Aldec HDL schematic and Verilog entry and simulation
- Altera Quartus synthesis and physical design tools
- Matlab
- Atmel WinAVR design tools
- Atmel Studio 4
- Android and Linux development tools

The hardware labs are supported by a Hardware Lab Manager, Bruce Hemingway, who orders parts and equipment as needed, organizes how the space is used, maintains parts and supplies used by the classes, and supervises a set of student lab assistants who provide support for the course labs. Bruce devotes half-time to lab management and half-time to teaching.

Software packages

Computing courses use a variety of Windows and Unix-based software tools, including:

- Java JDK
- Eclipse IDE
- SVN, Hg, and Git version control clients

- Squeak Smalltalk, Standard ML, Ruby, Perl, Python, CGI, PHP
- Bugzilla
- Android SDK
- MySQL, MSSQL, IIS, PostgreSQL
- VMWare and Virtual PC
- Adobe Master Collection
- Autodesk Education Suite for Entertainment Creation 2010, Pixar rendering tools
- Maya, AlfredForMaya

The department's computing and networking infrastructure is supported and maintained by a staff of 14 computing professionals. Campus networking is supported by the University's Computing and Communications staff.

B. Computing Resources

Departmental computing resources were described in the previous section. After those facilities, the next facility most commonly used by our students is their own machine plus publicly available cloud services. The university provides general purpose computing labs on campus, but use by our students is negligible.

C. Guidance

Use of computing is inherent in all our courses, and students acquire needed skills as part of doing course work. Our tech staff provides general help. A help desk is open every day, and additional help is available online. Students receive an overview of the lab facilities and policies during orientation to the department. In addition, we have extensive webpages that outline UW and departmental policies for use of computing resources

<http://www.cs.washington.edu/lab/policies-and-guidelines/> and that give instructions on how to use the resources available to them. Students can contact the CSE Support Office directly (by email or in person during business hours) for assistance with questions.

D. Maintenance and Upgrading of Facilities

Instructional computing equipment is monitored by an hourly 'ping' utility that gauges relative health of all PCs in the laboratories. As well, centrally reporting boot time scripts on each workstation log hardware components and patch/version level of software. Printers are routinely checked (multiple times per week) to ensure they are in working order and that supplies are stocked. Instructional laboratories are checked weekly for cleanliness. For Windows software images, machines images are "frozen" and reset all file modifications to original settings after a student logs out, to ensure uptime and student privacy between user sessions. Software updates

and security patches are pushed weekly during a staggered 'Maintenance Hour'. For Linux software, package updates and patches are pushed centrally, on-demand by tech staff. Major refresh/upgrades to instructional software packages most often are saved for quarter breaks. Computing hardware and facility upgrade opportunities are explored at least once a year

E. Library Services

The Engineering Library is a great resource for our faculty and staff. The Collections and Resources available are listed here: <http://www.lib.washington.edu/engineering/resources> and include everything from ACM Depository Collection to conference proceedings to technical reports. Our faculty and students have access not only to UW materials, but they can order from any organization within the Interlibrary Borrowing program. There is also a dedicated librarian to answer discipline-specific questions.

F. Overall Comments on Facilities

Most of our equipment is generic computing. There is some electronics work, but it is primarily low voltage. In the rare case that a student might encounter a piece of equipment that could cause injury, there is tight supervision by course staff.

CRITERION 8. INSTITUTIONAL SUPPORT

A. Leadership

The department leadership consists of the Chair, two Associate Chairs (one focusing on educational activities and the other focusing on development and external relations), and an Executive Committee consisting of these three individuals plus four elected members.

The Chair is appointed by the Dean for a five-year term; Levy was appointed in 2006 and recently agreed to a three-year extension, which ends in September 2014. The Associate Chairs are appointed by the Chair for short terms (one or more years) and for duties that correspond with current departmental needs and individual interests. Additional members of the Executive Committee are elected for one-year terms with a one-year reprieve after two consecutive terms.

The general duties of the Executive Committee are to be “in the loop” on all issues, to deal with straightforward issues without engaging the department as a whole, to approve simple appointments (e.g., Affiliate Faculty and postdocs), to ensure that the department as a whole is engaged on important issues, and to serve as a two-way communication channel.

We formed our first external advisory committee in 1999, but the committee had not met in many years. In 2008 we formed a new committee consisting of Barbara Liskov (chair), Randy Bryant, Deborah Estrin, Tom Mitchel, Christos Papadimitriou, Pat Hanrahan, Eric Horvitz, and Jeff Dean. That committee has given feedback on many department activities, including the recent curriculum revision, as explained in section 5.

B. Program Budget and Financial Support

The Department of Computer Science and Engineering has a strong foundation of funding provided by the State of Washington. In July 2012 the University increased state funding by \$1.6 million to allow for future growth. The annual budget for the current fiscal year (FY13) including salaries and operations is \$12 million. The majority of our state funding supports personnel including 46 tenured and tenure-track faculty and 6.5 instructors. State funding supports 7 advising staff 15 technical staff and administrative staff including the Assistant to the Chair, External Relations Manager, Receptionist, Facilities Manager, Events Coordinator, Administrator and 3.5 fiscal staff. State funding is augmented from several sources including CSE’s industrial affiliates program, corporate and individual gifts, and indirect cost returns (to the department) resulting from faculty awarded grants and contracts. These funds provide additional staff, a strong infrastructure and state-of-the-art facilities and equipment. Students do not have to manage machines themselves and can rely on a highly available and robust infrastructure for all their work and communications.

CSE receives funds through corporate and individual gifts and endowments. Continued fundraising campaigns have significantly increased the number of CSE’s endowed

professorships, chairs, fellowships and scholarships. CSE has 25 endowed scholarships, 15 endowed fellowships and 16 endowed professorships/chairs. Annually over 50 undergraduate scholarships are awarded. The Innovation Endowment generates \$450,000 annually in discretionary funds. Over 75 companies contribute to CSE's industrial Affiliates Program annually.

The UW Student Technology Fee Committee awarded \$139K in FY11 and \$78K in FY13 for improvements in instructional labs. The department also receives matching support in faculty recruiting from the College of Engineering and the Office of the Provost.

Over \$1.2 million in state funding is provided to support Teaching Assistants. CSE provides opportunities for over 80 undergraduate students to TA courses (50 support the introductory programming courses and 30 support other CSE courses). Approximately 30 graduate TAs support other CSE courses.

Computer Science and Engineering is in a strong fiscal position and resources are more than adequate to provide quality and excellence in our undergraduate program.

C. Staffing

CSE has 30 administrative and 50 technical/research staff. The average length of employment for the core technical staff is 17 years. The average length of employment for advisors and senior administrative staff positions is approximately 10 years, but many have been in CSE for 15 years or more. CSE is a supportive environment for staff, resulting in a high retention rate. Senior staff mentor and provide training for new staff hires. Staff development through attending conferences and training classes is encouraged and supported, both financially and with release time.

Staff Retention

1) Currently the UW is under salary and wage increase restrictions. The College of Engineering implemented a critical retention process for staff when we have or can reasonably anticipate difficulty retaining such employees. Each department nominated the candidate whom we anticipated as a retention risk.

2) At the College of Engineering we have offered staff opportunities to telework which enhances staff satisfaction and productivity ultimately impacting morale and retention in the midst of wage restrictions.

D. Faculty Hiring and Retention

Faculty hiring is managed through a Faculty Recruiting Committee (FRC), appointed by the chair. FRC is responsible for writing and publishing advertisements for open positions, evaluating applicants in collaboration with faculty in specific areas, and ultimately choosing the slate of candidates to be invited for interviews. Hiring decisions are made by the faculty as a

whole through a set of faculty meetings held all throughout the recruiting season. Final offers are chosen through a vote of the entire faculty. Last year (2012) the department hired 6 new faculty members – three senior faculty members at the associate professor level and three junior faculty members at the assistant level. The senior hires were national leaders in machine learning and data visualization (“big data”). The junior hires were experts in computer vision and wireless networks. This year we have three open searches: one joint with Electrical Engineering for a candidate in computer engineering, one joint with Statistics for a candidate in some aspect of data-driven discovery, and one for core CSE. There are three search committees, one for each search.

The department has a highly collegial and collaborative work environment, which along with outstanding students at all levels, provides an exciting and enjoyable work atmosphere. Our positive culture is well known nationally and helps to attract and retain faculty. The department supports its faculty by providing outstanding staff for grant administration and technical support. Seattle is a highly desirable location given the technology ecosystem surrounding the department. We have close ties to Microsoft Research, Google Seattle, and many other technical companies in the area; this provides opportunities for faculty and students for industrial collaboration, summer or sabbatical visits, student support and internships, etc. Proactive retention raises and other salary-increasing mechanisms (such as “A/B salary” – see below) have helped to increase salaries even during difficult periods of low state funding.

D. Strategies to Retain Current Qualified Faculty

In addition to the CSE retention efforts stated above, the College of Engineering uses the following to retain current qualified faculty members by:

- A. Implementing Retention Salary Adjustments – The Dean may request retention salary adjustments for qualified faculty through the Office of the Provost. Retention salary adjustments receive case-by-case review by the Office of the Provost, and additional documentation may be required such as a current curriculum vitae or case specific details. As a general principle, retention salary adjustments are expected to provide a minimum 5% salary increase. Generally, an individual may not receive a retention salary adjustment for a period of three years from the effective date of the most recent retention adjustment.
- B. Making opportunities available for A/B Retention Salary Adjustment - The fundamental purpose of the A/B Salary Policy for Faculty Retention is to insure that sufficient mechanisms exist to support the retention of UW tenured and tenure-track faculty consistent with the University of Washington Faculty Salary Policy. An A/B salary is comprised of an annual base salary with an A salary component and a B salary component. The A component is the state-committed salary support associated with tenure that is matched with an institutional expectation of teaching, research, and service contributions. The B component is the balance of the base salary funded from non-state

appropriated sources (e.g., grants, contract, and self-sustaining). The B component is contingent upon the faculty member's ability to generate funding from grants, contracts, or other applicable sources.

E. Support of Faculty Professional Development

In CSE, sabbaticals are permitted every 7 years and are supported by providing 2/3s salary over three quarters. The department has very liberal policies with regard to faculty leaves to spend time in industry or to spinout department-developed technology (e.g., in startups). In general, we support faculty members' requests for industrial interaction because we believe it enhances both our educational and research missions, creates new connections to industry, and gives faculty insight into important current problems faced by industry. Our Industrial Affiliates Program provides connections with over 100 high-tech companies both locally and nationally.

CSE provides support for staff members, instructors, and students to travel to workshops, seminars, and meetings. We also support the creation of campus-wide technical communities in areas of importance to us.

The University of Washington and the College of Engineering have extensive faculty professional development programs. Many of them focus on new faculty but some are for all faculty.

The University of Washington's Faculty Fellows Program orients new faculty to the University campus community. The Program is facilitated by a number of campus educators, including those that have received campus-wide teaching awards. Presenters and facilitators actively engage our new faculty members on a number of topics including, but not limited to, panel discussions with UW students, effective teaching methods and techniques for balancing the demands of successful teaching and research.

The University of Washington's Royalty Research Fund awards grants to faculty of up to \$40,000 to advance new directions in research, in particular:

- in disciplines for which external funding opportunities are minimal;
- for faculty who are junior in rank;
- in cases where funding may provide unique opportunities to increase applicant's competitiveness for subsequent funding.

Funded projects often lead to new creative activities or scholarly understandings, new scholarly materials or resources, and significant data or information that increase a faculty member's chances of obtaining new external funding.

The University of Washington Provost's Office provides bridge funding to support faculty to span the gap in critical research programs.

Faculty can receive up to \$50,000 (with a required 1:1 match from the department or college, meaning up to \$100,000) to help them maintain research productivity while they seek to obtain external funding for their labs.

There are a number of additional faculty professional development programs run by the College of Engineering:

1) Center for Engineering Learning & Teaching (CELT)

CELT supports the University of Washington, College of Engineering's mission by taking a leadership role in developing and supporting engineering instructional excellence. The CELT faculty development program employs an integrated multi-pronged agenda for improving engineering learning and teaching, which includes working with individual faculty members, conducting teaching workshops and seminars, providing teaching resource materials, and active participation in strategic-level initiatives. The CELT approach to professional faculty development begins with meeting and resolving the immediate concerns of faculty members. Simultaneously CELT helps faculty members place their improvement efforts within a larger cycle of ongoing improvement, implementation, and assessment. Workshop topics and specific instructional development activities and resources are identified through close cooperation with engineering faculty members. CELT services are available to all faculty members in the College of Engineering. For more information on CELT services see the CELT description in Table D-4 *Non-academic Support Units*.

2) ADVANCE: Increasing the Participation and Advancement of Women in Academic Science and Engineering Careers

The University of Washington received a \$3,750,000 National Science Foundation (NSF) ADVANCE Institutional Transformation grant in 2001 to increase the participation and advancement of women in academic science and engineering careers. With the grant, it formed the ADVANCE Center for Institutional Change (CIC) which is housed in the College of Engineering. The vision of the CIC is a campus in which all science, technology, engineering, and mathematics (STEM) departments are thriving, all faculty are properly mentored, and every STEM faculty member is achieving his or her maximum potential. UW believes that cultural changes that are designed to help underrepresented groups invariably help all groups and improve the environment for everyone.

The CIC implements programs designed to eliminate existing barriers and to precipitate cultural change at both the departmental and the institutional level. One of the successful strategies the ADVANCE program has employed to impact departmental culture and climate are quarterly leadership workshops for department chairs, deans, and emerging leaders. Prior to ADVANCE, department chairs received little or no professional development beyond their initial orientation

to the department chair position. The ADVANCE workshops provide those in leadership positions with a better understanding of the structural, psychological, and behavioral barriers to the advancement of faculty. For each half-day workshop, the department chairs are encouraged to invite an emerging leader so that other faculty can be exposed to academic leadership issues. These workshops have served as a forum for cross-college networking and professional development, and are the only regular department chair professional development gathering on campus. These workshops help develop the next set of department chairs in STEM departments. Department chairs have stated these workshops are the “boot camp” they never got and evaluations of the workshops have been uniformly high.

The UW ADVANCE program has had great impact. ASEE 2011 reports UW has 20.6% women faculty in engineering compared to a national average of 13.9%. Further, UW ranks seventh for number of women faculty in engineering, but institutions one through six have 38-73% more faculty in their colleges. Figure E-1 shows the percentage of women faculty in Engineering, compared to the national average.

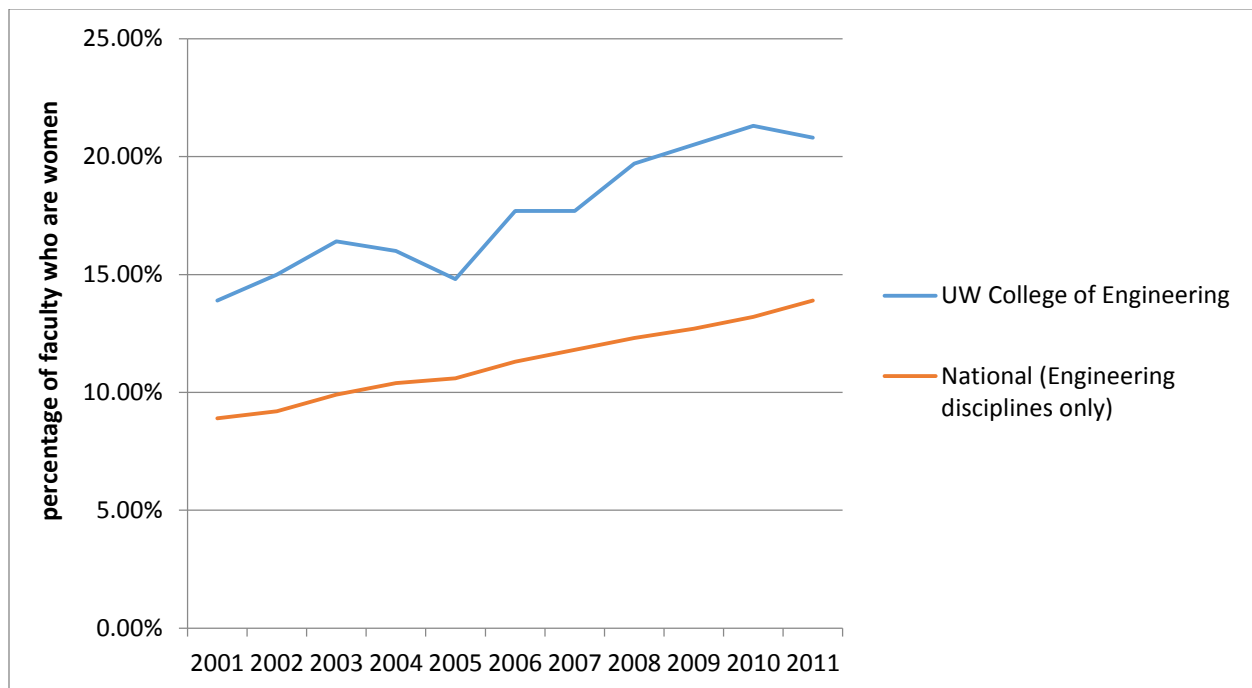


Figure E-1: Ladder Faculty Profile by Gender

The ADVANCE CIC also hosts a quarterly Pre-Tenure Faculty Workshop Series for all junior faculty members in STEM fields. This program provides participants the resources to establish a strong academic foundation and navigate the tenure process. Past topics have included the following: managing time and resources; understanding the tenure and promotion process; recruiting and mentoring graduate students; setting up a lab; building strong mentoring relationships; establishing peer networks and support structures; and applying for prestigious national grants, including the National Science Foundation's CAREER award. The workshops have been attended by 177 unique faculty members since 2005.

- 3) The College provides funds for junior faculty or faculty who have been out of research for some time to meet with program managers in Washington D.C. Developing and managing relationships with program managers is an important element in successful grantsmanship.

PROGRAM CRITERIA

The curriculum is structured to provide both depth and breadth. Students majoring in Computer Engineering have the some flexibility to tailor the program toward different career specializations. For example, we offer capstone design courses in topics as diverse as robotics, computational biology, embedded systems, and home networking. We do not require any single such course; instead we ensure that all of them fulfill broad curriculum needs, such as training in communication and extensive design and team work, and then allow students to choose.

A set of common core courses forms a foundation on which senior requirements are layered and appropriate capstone design courses provide an in-depth design experience. Our courses are structured to ensure that all students can achieve all the outcomes that we have defined for the program.

Our curriculum requirements are grouped into 9 categories:

- General education
- Visual, literary, and performing arts/Individuals & societies
- Written and oral communication
- Mathematics and statistics
- Science
- CSE core
- CSE senior electives
- Free electives

General education and VLPA/IS³ requirements expose our students to a variety of topics in the humanities and the natural world. Written and oral communication requirements include two technical writing courses⁴ in addition to basic English composition. Mathematics includes one-year of calculus and a linear algebra class. (We now teach a required probability/statistics class as CSE 312.) The science requirement includes two courses of Physics and 10 more credits of approved Natural Science. Students further increase their range by taking an additional three to six credits of approved natural science or higher-level math/statistics to bring their credit total to 45. They also all take CSE 311 and 312 where we count two credits each towards math. Free electives provide opportunities for students to expand their learning in any given area. A number of our students pursue a foreign language or a double major using this mechanism. Many use

³ Visual, Literary, and Performing Arts / Individual and Societies

⁴ The manner in which training in writing will be provided is being reviewed, as the College is terminating one of the writing courses we have relied on. We have already experimentally offered an in-department replacement, CSE 490W.

these credits for additional research credit in the department (6 credits of which can be counted in CSE electives, 9 credits if part of the Department or College Honors program) or to take more electives within the department.

Figure 5-1 – Required Course Pre-requisite Structure

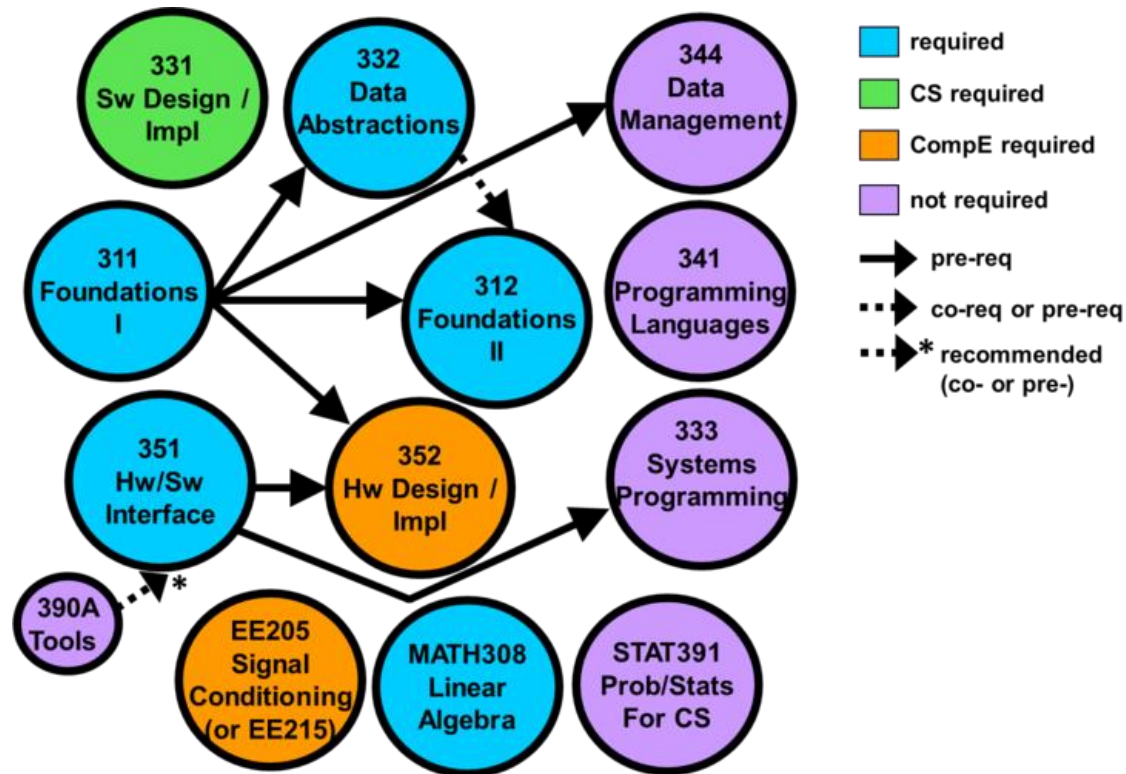


Figure 5-1 shows the pre-requisite structure of our 300-level courses. The heart of our curricular requirements is the departmental core requirements which are arranged in three tiers. First, students in both Computer Engineering and Computer Science take common core requirements that ensure a solid foundation in the discipline. This component consists of the same 4 courses for both majors (CSE 311, 312, 332, and 351). Computer engineering students also take 2 additional foundation courses: EE 205/215 (Introduction to Electrical Engineering/Introduction to Signal Conditioning) and CSE 352 (Hardware Design & Implementation).

The list of CSE core courses can be found at:

http://www.cs.washington.edu/students/ugrad/core_courses/.

Senior electives can be found at: http://www.cs.washington.edu/students/ugrad/cse_electives/.

The Computer Engineering curriculum is described in detail on the department undergraduate program site http://www.cs.washington.edu/students/ugrad/degree_requirements/.

In summary, the program consists of the following requirements based on 180 credits taken under a quarter system:

Core requirements (33 credits): CSE 142, 143, 311, 312, 332, 351, 352, and EE 215/205.

CE Senior Electives (39 credits)

- Four courses chosen from CSE 401, 403, 444, 451, 461, 466, 467, 471, and 484; at least one of which must be CSE 403, 466, or 484.
- Two additional courses from the CSE Core Course list on the CSE site:
http://www.cs.washington.edu/students/ugrad/core_courses/.
- One design capstone course from the approved list on the CSE site:
<http://www.cs.washington.edu/students/ugrad/capstone/>
- 10-12 additional credits from the CSE Electives list, including at least 7 credits from College of Engineering courses, to bring the total to 39 credits.

Math and Natural Sciences (45 credits)

- MATH (18 credits): Math124, 125, 126 and 308
- Science (20 credits): PHYS 121, 122; 10 credits of Natural Sciences
- CSE 311 (2 credits apply to math)
- CSE 312 (2 credits apply to math)
- Additional Math/Science Credits (3-6 to bring total to 45): Courses chosen from the approved natural science list on the CSE site:
http://www.cs.washington.edu/students/ugrad/natural_science/ or Stat 390, 391, 394, Math 307, 309, 334, 335, and AMATH 351, 353. (Stat 391 recommended.)

VLPA and I&S (30 credits): from UW provided list.

Written and Oral Communication (12 credits): English Composition, HCDE 231, 333

Free Electives (20-25 credits; depending on courses chosen)

APPENDICES

Appendix A – Course Syllabi

Written & Oral Communication

HCDE 231 – Introduction to Technical Writing

Credits: 3 credits (7 contact hours: 3 lecture/1 tutorial/3 lab)

Textbook/Materials

- *Technical Communication*, custom edition, available for purchase at the University Book Store. Look for Pearson (the publisher) on the binding.
- Purdue University's Online Writing Lab (OWL) – <http://owl.english.purdue.edu/>

Catalog Description: Reviews the fundamentals of writing, designing, and conveying technical information to various audiences. Using a process-centered approach, explores technical communication conventions such as organization, style, tone, illustration, and layout focusing on audience, purpose, and use to design and construct a variety of documents for academic and professional settings. Required of all engineering majors. Prerequisite: either C LIT 240, both ENGL 109 and ENGL 110, ENGL 111, ENGL 121, ENGL 131, ENGL 182, ENGL 197, ENGL 198, ENGL 199, or ENGL 281. Offered: AWSpS.

Required: Yes

Coordination: Human Centered Design and Engineering

Learning Objectives/Content:

This course introduces engineering undergraduates to the fundamental technical communication processes associated with writing, speaking, and teamwork. Specifically, you will learn:

- (1) about the interplay between audience, context, and purpose as you navigate the writing process to produce a variety of technical documents,
- (2) how to prepare and deliver formal and informal oral presentations, and
- (3) how to maximize the team process to conduct research and collaboratively produce a research report.

Engineers must attend to the ethical ramifications of both their engineering work and their professional communication. As a result, ethics will be a primary focus of this course and will be a specific component in several assignments.

Upon completion of this course, you will be able to:

Analyze a rhetorical situation and identify appropriate strategies based on a document's or presentation's purpose, audience, and context.

Recognize the basic features of technical writing genres and write within genre conventions.

Integrate text and visuals to clearly convey complex, technical information.

Revise documents for content, organization, and writing style.

Provide feedback to others on their writing, speaking, and teamwork abilities.

Demonstrate a professional style of working in teams and managing team projects.

Perform professional formal and informal presentation skills.

Be aware of ethical issues in engineering practice, team communication, and technical communication.

Demonstrate library research skills and appropriate source citation.

Mathematics & Natural Sciences

CHEM 142 General Chemistry

Credits: 5 credits (7 contact hours: 3 lecture/1 tutorial/3 lab)

Textbook/Materials

- ***Chemical Principles*, 7th ed.**, Steven Zumdahl (custom-split Chem 142 version contains Chapters 1-8 and the complete Student Solutions Manual).
- ***Study Guide, Chemical Principles*, 7th ed.**, Zumdahl/Kelter (*optional*).
- ***UW General Chemistry 142 Laboratory Manual, Autumn 2012-Summer 2013*** (Hayden McNeil)
- **UW Chemistry Laboratory Notebook** with numbered pages and carbonless duplicate pages. (Hayden McNeil).
- **ALEKS access.** Purchase online at: www.aleks.com (*see ALEKS document and links on the front page of the course website for more information and instructions for registering*).

Catalog Description: For science and engineering majors. Atomic nature of matter, stoichiometry, gas laws, chemical equilibrium, solubility, and acids and bases. Includes laboratory. Recommended: high school chemistry; placement into MATH 120 or higher. No more than the number of credits indicated can be counted toward graduation from the following course groups: CHEM 142, CHEM 145 (5 credits). Cannot be taken for credit if CHEM 120 already taken. Offered: AWSpS.

Required: No

Coordination: Department of Chemistry

Learning Objectives:

1. Learn to clearly define a problem and develop solutions for that problem including the use of central and auxiliary equations and conversion factors.
2. Learn to acquire and analyze data and correctly report experimental results (e.g., using an appropriate number of significant figures) in solutions to problems.
3. Develop a detailed understanding of the following fundamental chemistry topics:
 - The atomic nature of matter
 - Stoichiometry
 - Gases

- Chemical equilibrium
 - Applications of aqueous equilibria to acid/base and solubility chemistry
4. Conduct laboratory exercises that:
- Explore the concepts introduced in lectures.
 - Develop laboratory, data analysis, and scientific communication skills.

CHEM 152 General Chemistry

Credits: 5 credits (7 contact hours: 3 lecture/1 tutorial/3 lab)

Textbook, Supplemental Materials:

- *Chemical Principles*, 7th ed., Steven Zumdahl (custom-split Chem 152 version contains Chapters 8-12 and the complete Student Solutions Manual).
- *Study Guide, Chemical Principles*, 7th ed., Zumdahl/Kelter (*optional*).
- *UW General Chemistry 152 Laboratory Manual, Autumn 2012-Summer 2013* (Hayden McNeil)
- **UW Chemistry Laboratory Notebook** (Hayden McNeil) with numbered pages and carbonless duplicate pages. You may continue to use a notebook from a previous quarter if it meets the stated criteria and has at least 30 pages available.

Catalog Description: Energy, enthalpy and thermochemistry, spontaneity, entropy and free energy, electrochemistry, quantum mechanics, and atomic theory. Includes laboratory. Prerequisite: a minimum grade of 1.7 in either CHEM 142, or CHEM 145. No more than the number of credits indicated can be counted toward graduation from the following course groups: CHEM 152, CHEM 155 (5 credits). Offered: AWSpS.

Coordination: Department of Chemistry

Required: No

Learning Objectives/Course Topics

Applications of Aqueous Equilibria. We will begin the course by revisiting “equilibrium” introduced in Chem 142, specifically looking at acid/base systems involving buffers and titrations.

Thermodynamics. We will then explore the concepts of energy, enthalpy, entropy, and the Gibbs energy. The ultimate goal of this section of the course is to develop the tools that allow one to predict if a chemical reaction will be spontaneous.

Electrochemistry. We will revisit “redox” chemistry previously described in CHEM 142, and use this reaction class as an opportunity to explore the thermodynamic concepts described above.

Quantum Mechanics and Atomic Theory. In this portion of the course you will be introduced to the modern description of the hydrogen atom as derived from quantum mechanical principles. This description will be extended to other atoms, and eventually used to construct the periodic table of the elements.

In the **lab** portion of the course you will conduct laboratory exercises that:
Explore the concepts learned in lectures. Develop laboratory, data analysis, and scientific writing skills.

CHEM 162 General Chemistry

Credits: 5 credits (7 contact hours: 3 lecture/1 tutorial/3 lab)

Textbook/Supporting Materials

- ☐ *Chemical Principles*, 7th ed., Steven Zumdahl (custom-split Chem 162 version contains Chapters 13-21 and the complete Student Solutions Manual); other editions of the Chemical Principles text are okay to use.
- ☐ *Study Guide, Chemical Principles*, 7th ed., Zumdahl/Kelter (*optional*).
- ☐ *UW General Chemistry 162 Laboratory Manual, Autumn 2012-Summer 2013* (Hayden McNeil)
- ☐ **UW Chemistry Laboratory Notebook** (Hayden McNeil) with numbered pages and carbonless duplicate pages. You may continue to use a notebook from a previous quarter if it meets the stated criteria and has at least 30 pages available.

Catalog Description: General bonding and molecular-orbital theory, chemical kinetics, liquids and solids, properties of solutions, the elements in groups 1A-4A, the elements in groups 5A-8A, transition metals and coordination chemistry, and organic chemistry. Includes laboratory. Prerequisite: a minimum grade of 1.7 in CHEM 152. No more than the number of credits indicated can be counted toward graduation from the following course groups: CHEM 162, CHEM 165 (5 credits). Offered: A WS pS.

Coordination: Department of Chemistry

Required: Selected Elective

Learning Objectives/Course Topics

The central ideas presented in this course are:

- covalent bonding: orbitals
- chemical kinetics
- liquids and solids: structure and intermolecular forces
- properties of solutions
- representative elements - group 1A - 8A
- transition metals and coordination chemistry
- organic and biochemical molecules

In the **lab** portion of the course you will conduct laboratory exercises that: Explore the concepts learned in lectures. Develop laboratory, data analysis, and scientific writing skills

MATH 124 Calculus with Analytic Geometry I

Credits: 5 Credits (3 hrs lecture, 2 hours quiz section)

Textbook: *Calculus*, by Stewart, Vol. 1

Coordination: Department of Mathematics

Catalog Description: First quarter in calculus of functions of a single variable. Emphasizes differential calculus. Emphasizes applications and problem solving using the tools of calculus. Prerequisite: 2.5 in MATH 120, a score of 154-163 on the MPT-AS placement test, or score of 2 on AP test. Offered: AWSpS.

Required: Yes

Course Topics:

Week	Topics and Textbook Sections
1:	<ul style="list-style-type: none">• Tangents to circles , Tangents and velocity• Sec. 2.2-Limits
2:	<ul style="list-style-type: none">• Sec 2.3-Calculating Limits , Sec.2.5-Continuity , Sec.2.6-Asymptotes
3:	<ul style="list-style-type: none">• Sec. 2.7-3.1-Derivatives
4:	<ul style="list-style-type: none">• Sec. 3.2-3.4-More derivative rules, Trig derivatives, Chain rule
5:	<ul style="list-style-type: none">• Sec. 3.4-More chain rule, Sec. 10.2-Derivatives and parameterized

curves	
6:	<ul style="list-style-type: none"> • Sec. 3.5-Implicit differentiation, Sec. 3.6-Logarithmic differentiation • Sec. 3.9-Related rates
7:	<ul style="list-style-type: none"> • Sec. 3.10-Linear approximation • Sec. 4.1-Basics on min and max values
8:	<ul style="list-style-type: none"> • Sec. 4.3-Derivatives and shape of a curve, Sec. 4.4-L'hospital's rule
9:	<ul style="list-style-type: none"> • Sec. 4.5-Curve sketching , Sec. 4.7-Optimization
10:	<ul style="list-style-type: none"> • Sec. 4.7-More Optimization

MATH 125 Calculus with Analytic Geometry II

Credits: 5 Credits (3 hrs lecture, 2 hours quiz section)

Textbook: *Calculus*, by Stewart, Vol. 1

Coordination: Department of Mathematics

Catalog Description: Second quarter in the calculus of functions of a single variable. Emphasizes integral calculus. Emphasizes applications and problem solving using the tools of calculus. Prerequisite: either 2.0 in MATH 124, score of 3 on AB advanced placement test, or score of 3 on BC advanced placement test. Offered: AWSpS.

Required: Yes

Course Topics:

Week	Topics and Textbook Sections
1	Anti-derivatives, Areas and Distances and the Definite Integral. (Sec. 4.9, 5.1, 5.2)
2	The Fundamental Theorem of Calculus, Indefinite Integrals and Total Change, the Technique of Substitution. (Sec. 5.3, 5.4, 5.5)
3	Areas between Curves, Computing Volume: Washers and Shells. (Sec. 6.1, 6.2, 6.3)
4	Applications: Work and Average Value of a Function. Midterm #1 (Sec. 6.4, 6.5)
5	Techniques of Integration: Integration by Parts, Trigonometric Integrals and Trigonometric Substitution. (Sec. 7.1, 7.2, 7.3)
6	More Techniques: Partial Fractions and Combining Techniques. Approximation of Integrals. (Sec. 7.4, 7.5, 7.7)
7	Improper Integrals and the Length of a Curve. (Sec. 7.8, 8.1)
8	More Applications: Center of Mass. Midterm #2 (Sec. 8.3)
9	Introduction to Differential Equations: Separable Equations and Exponential Growth and Decay. (Sec. 9.1, 9.3, 3.8, [9.4 optional])
10	Final Exam Review.

MATH 126 Calculus with Analytic Geometry III

Credits: 5 Credits (3 hrs lecture, 2 hours quiz section)

Textbook: *Calculus*, by Stewart, Vol. 2

Catalog Description: Third quarter in calculus sequence. Introduction to Taylor polynomials and Taylor series, vector geometry in three dimensions, introduction to multivariable differential calculus, double integrals in Cartesian and polar coordinates. Prerequisite: either 2.0 in MATH 125, 2.0 in MATH 145, 2.0 in MATH 146, score of 5 on AB advanced placement test, or score of 4 on BC advanced placement test. Offered: AWSpS.

Required: Yes

Coordination: Department of Mathematics

Course Topics:

1. Vectors
2. Surfaces in space given by single equations with x , y and z (possibly not all three)
3. Curves
4. Calculus on curves
5. Partial Derivatives
6. Multivariable Integration
7. Taylor Series

MATH 307 Introduction to Differential Equations

Credits: 3 Credits (3 hours lecture)

Prerequisite: Math 125; Math 126 strongly recommended.

Text: *Elementary Differential Equations and Boundary Value Problems* by Boyce-DiPrima (Custom 9th ed.);

Catalog Description: Introductory course in ordinary differential equations. Includes first- and second-order equations and Laplace transform. Prerequisite: either 2.0 in MATH 125 or 2.0 in MATH 145. Offered: AWSpS.

Coordination: Department of Mathematics

Required: No

Topics Covered:

1. Review of First Order Equations - 8 lectures
2. Second Order ODE's - 12 lectures
3. Laplace Transforms - 6 lectures

MATH 308 Matrix Algebra with Applications

Credits:	3 Credits (3 lecture contact hours)
Text:	<i>Introduction to Linear Algebra</i> by Johnson, Riess and Arnold, 5th ed.
Catalog Description:	Systems of linear equations, vector spaces, matrices, subspaces, orthogonality, least squares, eigenvalues, eigenvectors, applications. For students in engineering, mathematics, and the sciences. Credit allowed for only one of MATH 308 or MATH 318. Prerequisite: either 2.0 in MATH 126 or 2.0 in MATH 146. Offered: AWP.
Prerequisite:	Math 126
Coordination:	Department of Mathematics
Required:	Yes

Course Topics:

§1 - Matrices and Systems of Linear Equations (9 lectures):

§1.1-1.4: Gaussian Elimination (3 lectures)

§1.5,1.6: Matrix operations (2 lectures)

§1.7: Linear independence (1 lecture)

§1.9: Data Fitting (1 lecture)

§1.9: Matrix inverses (2 lectures)

§2 - \mathbf{R}^n (11 lectures):

§3.1-3.3: Subspaces (3 lectures)

§3.4-3.5: Bases and dimension (3 lectures)

§3.6-3.7: Orthogonal bases and linear transformations (3 lectures)

§3.8,3.9: Least Squares (2 lectures)

§3 - Eigenvalues and eigenvectors (6 lectures)

§4.1-4.3: Introduction to eigenvalues and determinants (2 lectures)

§4.4-4.5: Eigenvalues, characteristic polynomial, eigenspaces. (2 lectures)

§4.8: Applications (2 lectures)

MATH 309 Linear Analysis

Credits: 3 Credits (3 lecture contact hours)

Text: *Elementary Differential Equations and Boundary Value Problems* by Boyce-DiPrima (Custom 9th ed.);

Catalog Description: First order systems of linear differential equations, Fourier series and partial differential equations, and the phase plane.
Prerequisite: either 2.0 in MATH 307 and 2.0 in MATH 308, or 2.0 in MATH 307 and MATH 318, or 2.0 in MATH 136.
Offered: AWSpS.

Prerequisites: Math 126, 307, 308

Coordination: Department of Mathematics

Required: No

Topics Covered:

1. Solving linear systems of ODE's (10 lectures):
2. The phase plane and critical points (1+ lectures):
3. Fourier series and boundary value problems (14 lectures)

PHYS 121 Mechanics

Credits: 5 Credits (3 hrs lecture, 2 hours quiz section)

Textbooks: - Tipler & Mosca, *Physics for Scientists and Engineers*, UW Custom Edition, 2008.

- McDermott, Shaffer et. al., *Tutorials in Introductory Physics*, Updated Preliminary 2nd Edition 2011-2012

- *Physics 121: Mechanics Laboratory*, by the Faculty of the Department of Physics, Autumn 2012, Winter/Spring/Summer 2013, published by Hayden-McNeil (ISBN 978-0-7380-5042-3).

Catalog Description: Basic principles of mechanics and experiments in mechanics for physical science and engineering majors. Lecture tutorial and lab components must all be taken to receive credit. Credit is not given for both PHYS 114 and PHYS 121. Prerequisite: either MATH 124 or MATH 134, which may be taken concurrently; recommended: one year high school physics. Offered: AWSpS.

Required: Yes

Coordination: Department of Physics

Course Topics

Week/Dates	Lab	Title
1: Apr. 2-5		No lab meeting. Do Pre-Lab 1*
2: Apr. 9-12	1	One-dimensional Kinematics
3: Apr. 16-19	2	Free-fall and Projectile Motion
4: Apr. 23-26	3	One-dimensional Dynamics
5: Apr. 30 - May 3	4	Newton's Third Law and Tension
6: May 7-10	5	Work and Energy
7: May 14-17	6	Momentum and Collisions
8: May 21-24	7	Rotational Kinematics
9: May 28-31	8	Torque Balance and Rotational Dynamics
10: June 4-7		<u>Make-up week.</u> One missed lab only.

PHYS 122 Electromagnetism

Credits: 5 Credits (3 hrs lecture, 2 hours quiz section)

Textbooks: - Tipler & Mosca, *Physics for Scientists and Engineers*, UW Custom Edition, 2008.

- McDermott, Shaffer et. al., *Tutorials in Introductory Physics*, Updated Preliminary 2nd Edition 2011-2012

- *Physics 122: Mechanics Laboratory*, by the Faculty of the Department of Physics, Autumn 2012, Winter/Spring/Summer 2013, published by Hayden-McNeil (ISBN 978-0-7380-5042-3).

Catalog Description: Covers the basic principles of electromagnetism and experiments in these topics for physical science and engineering majors. Lecture tutorial and lab components must all be taken to receive credit. Credit is not given for both PHYS 115 and PHYS 122. Prerequisite: either MATH 125 or MATH 134, which may be taken concurrently; PHYS 121. Offered: AWSpS.

Required: Yes

Coordination: Department of Physics

Course Topics:

Week	Topics
1	Electrostatics.
2	Electric Fields
3	Electric Circuits 1: Current
4	Electric Circuits 2: Potential Difference
5	Capacitors & RC Circuits
6	Introduction to Electronic Devices
7	Magnetic Fields and Forces
8	Electromagnetic Induction

PHYS 123: Waves

Credits: 5 Credits (3 hrs lecture, 2 hours quiz section)

Textbooks: - Tipler & Mosca, *Physics for Scientists and Engineers*, UW Custom Edition, 2008.

- *Physics 123: Mechanics Laboratory*, by the Faculty of the Department of Physics, Autumn 2012, Winter/Spring/Summer 2013, published by Hayden-McNeil (ISBN 978-0-7380-5042-3).

Catalog Description: Explores electromagnetic waves, the mechanics of oscillatory motion, optics, waves in matter, and experiments in these topics for physical science and engineering majors. Lecture tutorial and lab components must all be taken to receive credit. Credit is not given for both PHYS 116 and PHYS 123. Prerequisite: either MATH 126 or MATH 134, which may be taken concurrently; PHYS 122. Offered: AWSpS..

Required: No

Coordination: Department of Physics

Course Topics:

week	lecture topic	tutorial
1	Harmonic Oscillator I	Superposition & Reflection
	Harmonic Oscillator II	
	Traveling Waves	
2	Waves, Barriers & Doppler	Reflection & Transmission
	Standing Waves & Beats	
3	EM Waves	Refraction of waves
4	Polarization & Spectra	EM Waves
	Reflection & Refraction	
	Mirrors, Lenses & Images	
5	More on Images	2 Source Interference
	Optical Devices	
	Phase & Two Source Interference	
6	Diffraction & Phasors	Wave Properties of Light
	Slits & Gratings	
7	Light Waves & Photons	Multiple Slit Interference

	Matter Waves & Wave Functions	
	Atomic Structure & Spectra	
8	Relativity I	Single Slit Diffraction
	Relativity II	
	Relativity III	
9	Astrophysics and Nuclear Physics	
10	Nuclear Phys II: Fission & Fusion	Simultaneity
	Applications of Modern Physics	

STAT/MATH 390: Probability and Statistics in Engineering and Science

Credits:	4 Credits (4 hrs lecture, 1 hour quiz section)
Textbook:	<i>Applied Statistics for Engineers and Scientists</i> (2nd Edition), by Devore and Farnum.
Catalog Description:	Concepts of probability and statistics. Conditional probability, independence, random variables, distribution functions. Descriptive statistics, transformations, sampling errors, confidence intervals, least squares and maximum likelihood. Exploratory data analysis and interactive computing. Students may receive credit for only one of MATH 390, STAT/ECON 481, and ECON 580. Prerequisite: either MATH 126 or MATH 136.
Required:	No
Coordination:	Department of Mathematics
Course Topics:	<ol style="list-style-type: none">1. Conditional Probability2. Random Variables3. Distribution Functions4. Sampling Errors5. Confidence Intervals6. Least Squares7. Maximum Likelihood

STAT 391: Probability and Statistics for Computer Science

Credits: 4 Credits (4 hrs lecture)

Textbook: *None.*

Catalog Description: Fundamentals of probability and statistics from the perspective of the computer scientist. Random variables, distributions and densities, conditional probability, independence. Maximum likelihood, density estimation, Markov chains, classification. Applications in computer science. Prerequisite: 2.5 in MATH 126; 2.5 in MATH 308; either CSE 326, CSE 332, CSE 373, CSE 417, or CSE 421.

Required: No

Coordination: Department of Statistics

Learning Objectives/Course Topics:

It is the purpose of this course to teach the fundamental techniques underlying the vast majority of these statistical methods. The course is geared toward a computer science and engineering audience which has been exposed to a first course in probability and statistics.

We will first review and deepen the statistics and probability foundation, especially in the areas of: probability distributions, interactions of two or more variables, likelihood, estimation, and what makes a good estimator. In the second part of the course we will study specific models and paradigms for unsupervised (e.g. density estimation and modeling) and supervised (e.g. prediction) learning from data.

The course in its entirety will have a very significant experimental and practical component. Properties of models and estimators will be verified by experiments, the more advanced data analyses will be carried on real data sets. The numerical and algorithmic aspects of modern statistical analysis in the real world will be discussed as well.

Required

CSE 142 Computer Programming I

Credits

4.0, (3 hrs lecture, 1 hr section)

Lead Instructor

Stuart Reges

Textbook

- *Building Java Programs*, Reges

Course Description

Basic programming-in-the-small abilities and concepts including procedural programming (methods, parameters, return values), basic control structures (sequence, if/else, for loop, while loop), file processing, arrays and an introduction to defining objects.

Prerequisites

None

CE Major Status

Required

Course Objectives

Students will master basic procedural programming constructs. They will learn to write and debug small programs (50-100 lines) using a full range of procedural techniques using a variety of input sources (console, file) and a variety of output destinations (console, file, graphical). Students will learn the design principles that are relevant to this style of “programming in the small” including decomposition, information hiding, elimination of redundancy, detailed documentation, and use of parameters and return values to create flexible components. Students will be exposed to the major design issues of object oriented programming including inheritance.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (e) an ability to identify, formulate, and solve engineering problems
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Basic concepts of computer structure and program execution

- Variables, types, expressions, and assignment
- Input/output: console, file, graphical
- Conditional execution (if/else)
- Iteration (for, while)
- Defining methods: parameters, return values
- Arrays (one-dimensional)
- User-defined classes
- Use of standard library objects: strings, graphics
- Procedural decomposition of problems
- Programming style: eliminating redundancy, localizing variables, class constants, commenting, use
- of parameters and return values to increase flexibility, appropriate choice of control structure
- (e.g., sequential if versus if/else)

CSE 143 Computer Programming II

Credits

5.0 (3 hrs lecture, 2 hrs section)

Lead Instructor

Stuart Reges

Textbook

- *Building Java Programs*, Reges

Course Description

Continuation of CSE 142. Concepts of data abstraction and encapsulation including stacks, queues, linked lists, binary trees, recursion, instruction to complexity and use of predefined collection classes.

Prerequisites

CSE 142.

CE Major Status

Required

Course Objectives

Students learn about data abstraction and the basic design principles of object oriented programming. Students will become familiar with standard data abstractions (lists, maps, sets, stacks, queues) as well as a variety of implementation techniques (arrays, linked lists, binary trees). Students will learn to program using recursion and recursive backtracking. Students will learn the object-oriented constructs that support code reuse (encapsulation, interfaces, inheritance, abstract classes) as well as learning how to make use of off-the-shelf components from libraries like the Java Collections Framework.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (e) an ability to identify, formulate, and solve engineering problems
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Abstract data types: stacks, queues, lists, maps, sets

- Implementing linked structures (linked lists, binary trees)
- Recursion and recursive backtracking
- Using off-the-shelf components (e.g., Java Collections Framework)
- Use of inheritance for additive change and factoring out common code into abstract classes
- Class design: encapsulation, documentation, throwing exceptions, appropriate choice of fields
- Thorough testing and debugging
- Time and space complexity
- Efficient sorting and searching algorithms (binary search, mergesort)
- Iterator use and implementation

CSE 311 Foundations of Computing I

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Paul Beame

Textbook

- *Discrete Math & Its Applications*, Rosen

Course Description

Examines fundamentals of logic, set theory, induction, and algebraic structures with applications to computing; finite state machines; and limits of computability.

Prerequisites

CSE 143; either MATH 126 or MATH 136.

CE Major Status

Required

Course Objectives

At the end of this course, students will be able to:

1. *express* simple mathematical concepts formally
2. *understand* formal logical expressions and *translate* between natural language expressions and predicate logic expressions
3. *manipulate and understand* modular arithmetic expressions
4. *create* simple proofs, including proofs by induction
5. *design* two-level logic circuits to compute Boolean functions
6. *design* simple finite state machines both with and without output
7. *design and interpret* regular expressions representing sets of strings
8. *recognize* that certain properties of programs are undecidable

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (e) an ability to identify, formulate, and solve engineering problems

Course Topics

- Propositional/Boolean logic (3-4 lecture hours)
- Predicate Logic (2 lecture hours)
- Logical Inference (2 lecture hours)
- Sets and Functions (0.5-1 lecture hour)
- Arithmetic (3-4 lecture hours)
- Mathematical Induction and Applications (5-6 lecture hours)
- Relations and Directed Graphs (1.5-2 lecture hours)
- Finite-State Machines (4.5-5 lecture hours)
- Circuits for finite state machines (1 lecture hour)
- Turing Machines and Undecidability (3-4 lecture hours)

CSE 312 Foundations of Computing II

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Martin Tompa

Textbook

- *Introduction to Probability*, Bertsekas & Tsitsiklis

Course Description

Examines fundamentals of enumeration and discrete probability; applications of randomness to computing; polynomial-time versus NP; and NP-completeness.

Prerequisites

CSE 311; CSE 332, which may be taken concurrently.

CE Major Status

Required

Course Objectives

Course goals include an appreciation and introductory understanding of (1) methods of counting and basic combinatorics, (2) the language of probability for expressing and analyzing randomness and uncertainty (3) properties of randomness and their application in designing and analyzing computational systems, (4) some basic methods of statistics and their use in a computer science & engineering context

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (e) an ability to identify, formulate, and solve engineering problems

Course Topics

- permutations, combinations
- pigeonhole principle
- inclusion-exclusion
- probability axioms
- conditional probability
- law of total probability

- Bayes' Rule
- independence
- random variables
- expectation and variance
- joint distributions
- binomial distribution, geometric distribution, Poisson distribution,
- continuous random variables
- uniform distribution, exponential distribution, normal distribution
- central limit theorem
- randomized algorithms
- Markov and Chebyshev inequalities
- Chernoff bounds
- law of large numbers
- maximum likelihood estimate

CSE 332 Data Abstractions

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Ruth Anderson

Textbook

- *Data Structures & Algorithm Analysis in Java*, Weiss

Course Description

Covers abstract data types and structures including dictionaries, balanced trees, hash tables, priority queues, and graphs; sorting; asymptotic analysis; fundamental graph algorithms including graph search, shortest path, and minimum spanning trees; concurrency and synchronization; and parallelism. Not available for credit for students who have completed CSE 373.

Prerequisites

either CSE 311 or CSE 321.

CE Major Status

Required

Course Objectives

1. *communicate* the representation and organization of data in terms of ubiquitous computing abstractions such as stacks, queues, trees, hash-tables, and graphs
2. *analyze* algorithms for correctness and efficiency, including the use of asymptotic analysis
3. *design* parallel programs that use extra computational resources to complete a task more quickly
4. *recognize* software errors related to concurrent execution of tasks such as race conditions
5. *create* software that implements classic data structures and algorithms and uses such algorithms appropriately

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic

constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(e) an ability to identify, formulate, and solve engineering problems

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Review of simple abstract data-types (2 lecture hours)
- Algorithm Analysis (1 lecture hour)
- Asymptotic Analysis (2 lecture hours)
- Priority Queues (2 lecture hours)
- Dictionaries (5 lecture hours)
- Sorting (3 lecture hours)
- Graphs (4 lecture hours)
- Parallel Programming (4 lecture hours)
- Concurrent Programming (3 lecture hours)

CSE 351 The Hardware/Software Interface

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Luis Ceze

Textbook

- *Computer Systems*, Bryant
- *C Programming Language*, Kernighan

Course Description

Examines key computational abstraction levels below modern high-level languages; number representation, assembly language, introduction to C, memory management, the operating-system process model, high-level machine architecture including the memory hierarchy, and how high-level languages are implemented.

Prerequisites

CSE 143.

CE Major Status

Required

Course Objectives

At the end of this course, students should:

1. understand the multi-step process by which a high-level program becomes a stream of instructions executed by a processor;
2. know what a pointer is and how to use it in manipulating complex data structures;
3. be facile enough with assembly programming (X86) to write simple pieces of code and understand how it maps to high-level languages (and vice-versa);
4. understand the basic organization and parameters of memory hierarchy and its importance for system performance;
5. be able to explain the role of an operating system;
6. know how Java fundamentally differs from C;
7. grasp what parallelism is and why it is important at the system level; and
8. be more effective programmers (more efficient at finding bugs, improved intuition about system performance).

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Number representation: Two's complement, signed vs. unsigned, floating point (1 week)
- Assembly (2 weeks)
 - Memory vs. registers
 - Instruction format
 - Control structures in assembly (loops, procedure calls)
- C (2 weeks)
 - Pointers, arrays, strings
 - Memory management, malloc/free, stack vs. heap
 - structs
- Compilation, linking, libraries (code across multiple files) (0.5 weeks)
- The process model (what the operating system provides, not how it provides it) (1 week)
 - Virtualization and isolation (including virtual memory)
 - Components of a process state and notion of a context switch
 - System calls for accessing shared resources and communication channels
 - Asynchronous signals
- High-level machine architecture (2 weeks)
 - Register file
 - Instruction cycle
 - Caching and the memory hierarchy
- The Java-to-C connection (1 week)
 - Representing an object as pointer to struct with pointer to method-table; performing a method call
 - Constructors as malloc-then-initialize
 - Garbage collection via reachability from the stack
 - Java array-bounds-checking via array-size fields
- Parallelism/multicore/threads (0.5 weeks)

CSE 352 Hardware Design and Implementation

Credits

4.0 (3 hrs lecture, 3 hr lab)

Lead Instructor

Mark Oskin

Textbook

- *Digital Design & Computer Architecture*, Harris

Course Description

Covers digital circuit design, processor design, and systems integration and embedded-systems issues. Includes substantial hardware laboratory.

Prerequisites

CSE 311; CSE 351.

CE Major Status

Required

Course Objectives

At the end of this course, students should:

1. know how to implement a Boolean function in hardware, and how to analyze the cost and performance of the implementation
2. understand system clocking methodology to implement sequential circuits
3. understand the timing constraints imposed by the clocking methodology and how to analyze a digital system for timing correctness
4. understand the basics of computer arithmetic
5. understand how to implement an instruction set processor using the digital design methodology
6. understand how to use pipelining to improve the performance of a digital circuit
7. understand how to use forwarding, stalling and prediction to address hazards in pipelined processors
8. be able to use design tools to design and implement digital circuits using FPGA technology
9. be able to write and debug assembly language programs, including the use of interrupts and timers for real-time operation

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve engineering problems
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Implementation of Boolean functions (6 lectures)
- Implementation of sequential circuits (6 lectures)
- FPGA architectures and CAD tools (2 lectures)
- Y86 processor design (8 lectures)
- Support for real-time, embedded systems (3 lectures)
- Pipelining (3 lectures)

EE 205 Introduction to Signal Conditioning

Credits: 4 (3 hours lecture; 1 hour lab)

Catalog Description

Introduction to analog circuits interfacing sensors to digital systems. Connection, attenuation, amplification, sampling, filtering, termination, controls. Kirchoff's Laws, sources, resistors, op amps, capacitors, inductors. PSpice, Matlab.

Prerequisites: either Math 126 or Math 136; PHYS 122. Intended for non-EE majors.

Coordinator: To be determined.

CE Major Status: one of EE 205 and EE 215 is required

Course Objectives

To introduce basic electrical engineering concepts used in the connection of sensors to digital systems, acting as a first course in Electrical Engineering for non-EE majors. To learn the "alphabet" of circuits, including wires, resistors, capacitors, inductors, independent and dependent voltage and current sources, and operational amplifiers. To prepare students to deal with sensor I/O in digital system courses.

1. At the end of this course, students will be able to:
Describe the role of signal conditioning in digital systems.
2. Design voltage divider circuits to attenuate voltage signals for digital input.
3. Design simple op amp circuits to amplify sensor outputs, including saturation and slew rate considerations.
4. Describe techniques to deal with noisy sensors.
5. Design simple high pass and low pass passive and active filters.
6. Acquire and analyze analog signals in Matlab.
7. Analyze simple circuits using PSpice.
8. Determine the sample rates necessary signals with specific frequency content and describe the effects of improper sample rates.
9. Design anti-aliasing filters.
10. Explain analog transmission line effects on digital signals.
11. Design terminations for digital signal lines.
12. Describe the importance of isolation.
13. Design isolation methods.
14. Sketch a simple control system block diagram and explain its basic operation.

Textbook

Horowitz and Hill, The Art of Electronics, 2nd Ed, Cambridge Press, 1989.
Also Hayes and Horowitz, Student Manual for the Art of Electronics, Cambridge Press, 1989.

Reference Texts

Pallàs-Areny and Webster, Sensors and Signal Conditioning, 2nd Edition, John Wiley & Sons, 2001.

Nilsson and Riedel, Electric Circuits, 8th Edition. Prentice Hall, 2008, or any other circuit theory textbook.

Topics

1. Sensors, signals, A/D conversion, signal conditioning paradigm. Voltage, current.
2. Attenuation, Ohm's Law, resistance, voltage divider, resistor precision, power dissipation.
3. Amplification, ideal op amps, inverting amplifier, non-inverting amplifier, saturation, slew rate, bandwidth, impedance matching.
4. Strain gauges, noise, frequency domain, filtering, FFT, low pass passive filter design, PSpice.
5. High pass passive filters. Phase angle. Active filters.
6. Aliasing, Nyquist limit, anti-aliasing filters. Sample and Hold.
7. Digital signals and transmission lines – attenuation, ringing, termination, cross talk.
8. Signal isolation. Protection.
9. Control systems. PID control.

ABET Outcomes

- (a) An ability to apply knowledge of math, science and engineering.
- (b) An ability to design and conduct experiments, as well as to analyze and interpret data.
- (c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
- (d) An ability to function on multidisciplinary teams.
- (e) An ability to identify, formulate and solve engineering problems.
- (g) An ability to communicate effectively.
- (k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

EE 215 Fundamentals of Electrical Engineering

Credits: 4 (3 hrs lecture, 1 hr section)

[UW Course Catalog Description:](#)

Coordinator: M.P. Anantram, Professor of Electrical Engineering

Goals: To develop the fundamental tools of linear circuit analysis which will be useful to all engineers. To learn the "alphabet" of circuits, including wires, resistors, capacitors, inductors, independent and dependent voltage and current sources, and operational amplifiers. To prepare students for more advanced courses in circuit analysis.

Learning Objectives: At the end of this course, students will be able to:

1. *Identify* linear systems and represent those systems in schematic form
2. *Explain* precisely what the fundamental circuit variables mean and why the fundamental laws governing them are true.
3. *Apply* Kirchhoff's current and voltage laws, Ohm's law, and the terminal relations describing inductive and capacitive energy-storage elements to circuit problems.
4. *Simplify* circuits using series and parallel equivalents and using Thevenin and Norton equivalents
5. *Perform* node and loop analyses and set these up in standard matrix format
6. *Explain* the physical underpinnings of capacitance and inductance.
7. *Identify and model* first and second order electric systems involving capacitors and inductors
8. *Predict* the transient behavior of first and second order circuits

Textbook: Nilsson and Riedel, *Electric Circuits, 8th Edition*. Prentice Hall, 2008

Reference Texts: none

Prerequisites by Topic:

1. Fundamental physics (PHYS 122), including concepts of power, energy, force, electric current, and electric fields
2. Fundamental mathematics (MATH 126), trigonometric and (complex) exponential functions, introductory differential and integral calculus, first and second order linear differential equations

Topics:

1. Fundamental electric circuit quantities (charge, current, voltage, energy, power) [0.5 week]
2. The "alphabet" of circuit schematics (resistors, wires, sources, etc.) [0.5 week]
3. Analysis, graph theory concepts: loops, nodes, supernodes [0.5 week]
4. Kirchhoff's current and voltage laws [0.5 week]
5. Ohm's law [0.5 week]
6. Series and parallel resistor combinations, voltage and current division [1 week]
7. Thevenin and Norton equivalents; linearity and superposition solution methods [1 week]
8. Linear algebraic techniques (node analysis; loop/mesh analysis) [2 weeks]
9. Op amp circuits [1 week]
10. Capacitors and inductors [0.5 week]
11. First and second order circuits in the time domain [2 weeks]

Outcome Coverage:

- (a) An ability to apply knowledge of math, science and engineering
- (b) An ability to design and conduct experiments, as well as to analyze and interpret data.
- (c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) An ability to function on multidisciplinary teams.
- (e) An ability to identify, formulate and solve engineering problems.

CE Senior Electives

CSE 331 Software Design and Implementation

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Michael Ernst

Textbook

- *Effective Java*, Bloch
- *Pragmatic Programmer*, Hunt & Thomas

Course Description

Explores concepts and techniques for design and construction of reliable and maintainable software systems in modern high-level languages; program structure and design; program-correctness approaches, including testing; and event-driven programming (e.g., graphical user interface). Includes substantial project and software-team experience.

Prerequisites

CSE 143.

CE Major Status

Selected Elective

Course Objectives

There is a level of programming maturity beyond introductory programming that comes from building larger systems and understanding how to specify them precisely, manage their complexity, and verify that they work as expected. After completing this course successfully students should be able to:

- Successfully build medium-scale software projects in principled ways
- Understand the role of specifications and abstractions and how to verify that an implementation is correct, including effective testing and verification strategies and use of formal reasoning
- Analyze a software development problem and be able to design effective program structures to solve it, including appropriate modularity, separation of abstraction and implementation concerns, use of standard design patterns to solve recurring design problems, and use of standard libraries

- Use modern programming languages effectively, including understanding type systems, objects and classes, modularity, notions of identity and equality, and proper use of exceptions and assertions
- Gain experience with contemporary software tools, including integrated development environments, test frameworks, debuggers, version control, and documentation processing tools

To gain experience we will use Java and associated tools like Eclipse, JUnit, JavaDoc, and Subversion, but the goal is to understand the underlying ideas and concepts that are widely applicable to software construction.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve engineering problems
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Reasoning about programs: pre- and post-conditions, invariants, and correctness
- Abstract data types, specification, implementation, abstraction functions, representation invariants, notions of equality
- Java language issues: subclasses and subtypes, generics, exceptions, assertions, etc.
- Tools: Eclipse IDE, version control, svn
- Code quality, style, comments, documentation, JavaDoc
- Testing, test coverage, black- and white-box testing, test-first development, regression testing, JUnit
- Debugging strategies and tools
- Design: modular design, coupling, cohesion; design patterns; basic UML as a design notion
- User interfaces, callbacks, separation of model from view/control

CSE 333 Systems Programming

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Steve Gribble

Textbook

- *C++ Primer*, Lippman

Course Description

Includes substantial programming experience in languages that expose machine characteristics and low-level data representation (e.g., C and C++); explicit memory management; interacting with operating-system services; and cache-aware programming.

Prerequisites

CSE 351.

CE Major Status

Selected Elective

Course Objectives

At the end of this course, students should:

1. have an in-depth understanding of the C and C++ programming languages and competence at programming to strict style guidelines;
2. understand manual memory management, pointer-based data structure construction and manipulation, and their debugging challenges;
3. learn the file-system and networking system call API and use it to construct systems such as an on-disk index and a client/server application;
4. be familiar with programming tools like debuggers, unit test frameworks, code coverage, profilers, and code review systems; and,
5. gain rudimentary familiarity with concurrent programming, including processes, threads, and event-driven code.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (c) an ability to design a system, component, or process to meet desired needs within realistic

constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(e) an ability to identify, formulate, and solve engineering problems

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- C programming (2 weeks)
 - pointers, structs, casts; arrays, strings
 - dynamic memory allocation
 - C preprocessors, multifile programs
 - core C libraries
 - error handling without exceptions
- C++ programming (4 weeks)
 - class definitions, constructors and destructors, copy constructors
 - dynamic memory allocation (new / delete), smart pointers, classes with dynamic data inheritance, overloading, overwriting, and dynamic dispatch
 - C++ templates and STL
- Tools and best practices (2 weeks)
 - compilers, debuggers, make
 - leak detectors, profilers and optimization, code coverage
 - version control
 - code style guidelines; code review
- Systems topics: the layers below (OS, compiler, network stack) (2 weeks)
 - concurrent programming, including threading and asynchronous I/O
 - fork / join, address spaces, the UNIX process model
 - file-system API
 - network sockets API
 - understanding the linker / loader

CSE 341 Programming Languages

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Daniel Grossman

Textbook

- Depends on instructor, often relying on free language user's guides, online tutorials, and/or instructor-written reading notes and/or videos

Course Description

Basic concepts of programming languages, including abstraction mechanisms, types, and scoping. Detailed study of several different programming paradigms, such as functional, object-oriented, and logic programming. No credit if CSE 413 has been taken.

Prerequisites

CSE 143

CE Major Status

Selected Elective

Course Objectives

- To understand fundamental programming-language concepts
- To become fluent in non-imperative programming paradigms
- To become able to learn new programming languages efficiently

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve engineering problems
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- The following topics are always covered:
 - functional programming (avoiding mutation; exploiting recursion and higher-order functions; closures; anonymous functions)
 - algebraic datatypes and pattern-matching
 - essential object-oriented programming (late-binding / dynamic dispatch, subtyping vs. subclassing)
 - language support for abstraction, such as modules, abstract types, and dynamic type-creation
 - syntax vs. semantics
 - static vs. dynamic typing
 - parametric polymorphism / generics
 - object-oriented extensibility vs. functional extensibility
- In any particular offering, most of the following are covered:
 - tail recursion
 - currying
 - equality vs. identity
 - macros
 - type inference
 - lazy evaluation and related idioms such as streams and memorization
 - code-as-data concepts, such as reflection and eval/apply
 - lexical vs. dynamic scope
 - subtyping issues such as structural vs. named subtyping and subsumption vs. coercion
 - object-oriented concepts such as multiple inheritance, multimethods, and metaclasses
 - bounded parametric polymorphism
 - forms of parameter passing
 - subtype polymorphism and bounded polymorphism
 - logic programming
- A small number of these topics might also be covered:
 - language-design principles
 - history of programming languages
 - programming environments
 - debugging support
 - compilers vs. interpreters
 - continuations
 - continuation-passing style
 - coroutines
 - iterators
 - language support for concurrency

CSE 344 Introduction to Data Management

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Dan Suciu

Textbook

- *Database Systems: The Complete Book*, Garcia-Molina

Course Description

Introduces database management systems and writing applications that use such systems; data models (e.g., relational, semi-structured), query languages (e.g., SQL, XQuery), language bindings, conceptual modeling, transactions, security, database tuning, data warehousing, parallelism, and web-data management.

Prerequisites

either CSE 311 or CSE 321.

CE Major Status

Selected Elective

Course Objectives

At the end of this course, students should:

1. Understand basic concepts related to data management such as the notion of data models, physical and logical data independence, query languages, query execution, and cost-based query optimization.
2. Acquire knowledge of the relational data model including the relational algebra, datalog and the relational calculus.
3. Acquire knowledge of the semi-structured data model.
4. Learn several declarative query languages such as SQL, XQuery, and Pig Latin
5. Know how to use a relational database management system and a semi-structured database management system. This includes designing a database, loading data into the database, updating the database, and querying the database.
6. Understand the notion of transaction processing, ACID properties, and different degrees of consistency.

7. Write applications that use a database management system as a back-end. Know how to use transactions when developing such applications.
8. Understand the principles of parallel data processing and know how to use a parallel relational database system or a MapReduce system.
9. Understand the concept of data management as a service in the cloud and gain basic experience with using data management systems hosted in a public cloud such as SQL Azure and Amazon Elastic MapReduce.
10. Understand challenges of data management related to data integration and data cleaning.
11. Gain experience managing a large dataset (e.g., 0.5 TB graph).

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Introduction to the notion of data models and the relational data model (2 lecture hours)
- The declarative query language SQL (4 lecture hours)
- Relational algebra, relational calculus, and datalog (3 lecture hours)
- Query evaluation and database tuning (2 lecture hours)
- The semi-structured data model: XML, XPath, and XQuery (2 lecture hours)
- Database design (4 lecture hours)
- Transaction processing (3 lecture hours)
- Parallel data processing (4 lecture hours)
- Data management as a service in the cloud (1 lecture hour)
- Data integration and data cleaning (2 lecture hours)

CSE 401 Introduction to Compiler Construction

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Dan Grossman

Textbook

- *Engineering a Compiler*, Cooper & Torczon

Course Description

Fundamentals of compilers and interpreters; symbol tables; lexical analysis, syntax analysis, semantic analysis, code generation, and optimizations for general purpose programming languages. No credit to students who have taken CSE 413.

Prerequisites

either CSE 326 and CSE 378 or CSE 332 and CSE 351.

CE Major Status

Selected Elective

Course Objectives

Learn principles and practice of language implementations. Understand tradeoffs between run-time and compile-time processing. Understand tradeoffs between language features, run-time efficiency, and implementation difficulty. Gain experience working with large systems software, object-oriented design, and Java.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve engineering problems
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Organization of Compilers and Interpreters

- Lexical Analysis
- Syntactic Analysis
- Semantic Analysis
- Interpretation
- Run-Time Storage Layout
- Code Generation
- Optimization

CSE 403 Software Engineering

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Michael Ernst

Textbook

None

Course Description

Fundamentals of software engineering using a group project as the basic vehicle. Topics covered include the software crisis, managing complexity, requirements specification, architectural and detailed design, testing and analysis, software process, and tools and environments.

Prerequisites

either CSE 303 or CSE 331; either CSE 332 or CSE 326; recommended: either CSE 331 or project experience in a work setting.

CE Major Status

Selected Elective

Course Objectives

A central objective of the course is to have students develop a deep understanding of the distinctions between software engineering and programming. In addition, the students understand the software lifecycle, increase their knowledge of classic and modern software engineering techniques, and develop concrete experience in turning ill-formed concepts into products working with a team.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global,

economic, environmental, and societal context

(i) a recognition of the need for, and an ability to engage in life-long learning

(j) knowledge of contemporary issues

Course Topics

- Topics covered include the software crisis, managing complexity, requirements specification, architectural and detailed design, testing and analysis, software process, and tools and environments.

CSE 421 Introduction to Algorithms

Credits

3.0 (3 hrs lecture)

Lead Instructor

Anup Rao

Textbook

- *Algorithm Design*, Kleinberg

Course Description

Techniques for design of efficient algorithms. Methods for showing lower bounds on computational complexity. Particular algorithms for sorting, searching, set manipulation, arithmetic, graph problems, pattern matching.

Prerequisites

either CSE 312 or CSE 322; either CSE 326 or CSE 332.

CE Major Status

Selected Elective

Course Objectives

Learn basic techniques for design and analysis of algorithms, including correctness proofs. Learn a number of important basic algorithms. Learn how to prove that problems are NP-complete.

ABET Outcomes

(a) an ability to apply knowledge of mathematics, science, and engineering

Course Topics

- Main Techniques:
 - Design: Induction, Graph search, Divide and Conquer, Greedy, Dynamic Programming, Network Flow
 - Analysis: Asymptotic Analysis, Recurrences.
 - Intractability: Reduction.
- Typical Algorithm coverage:
 - depth- and breadth-first search
 - bi- and/or strongly connected components
 - shortest paths

- min spanning trees
 - transitive closure
 - flows and matchings
 - Strassen's method
 - FFT
 - knapsack
 - edit distance/string matching
 - scheduling
- Intractability:
 - reduction
 - P / NP
 - verification/certificates/witnesses,
 - nondeterminism,
 - completeness
 - Example problems:
 - SAT
 - 3-SAT
 - clique
 - vertex cover
 - 0-1 knapsack
 - partition
 - coloring

CSE 427 Computational Biology

Credits

3.0 (3 hrs. lecture)

Lead Instructor

Martin Tompa

Textbook

None

Course Description

Algorithmic and analytic techniques underlying analysis of large-scale biological data sets such as DNA, RNA, and protein sequences or structures, expression and proteomic profiling. Hands-on experience with databases, analysis tools, and genome markers. Applications such as sequence alignment, BLAST, phylogenetics, and Markov models.

Prerequisites

either CSE 326 or CSE 332

CE Major Status

Selected Elective

Course Objectives

The goal of this course is to give students enough exposure to Computational Biology that they can make informed choices about further exploration such as graduate programs. Another goal is to give the students hands-on experience with databases, analysis tools, and genome browsers.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (j) knowledge of contemporary issues

Course Topics

1. Basics of Molecular Biology: proteins, DNA, RNA; DNA replication, transcription, translation, regulation of expression; gene structure; genome organization
2. Pairwise sequence alignment: optimal global and local alignment, affine gap penaltie

3. The algorithms and statistics of BLAST
4. Amino acid substitution matrix
5. Multiple sequence alignment: definition and NP-completeness, comparative sequence analysis and the UCSC Human Genome Browser, progressive alignment, methods for whole-genome multiple sequence alignment
6. Inference of phylogenetic trees
7. Markov chains and hidden Markov models
8. Challenges in Computational Molecular Biology

CSE 428 Computational Biology Capstone

Credits

5.0 (5 hrs. lecture/meeting times)

Lead Instructor

Martin Tompa

Textbook

None

Course Description

Designs and implements a software tool or software analysis for an important problem in computational molecular biology.

Prerequisites

either CSE 303 or CSE 331; either CSE 326 or CSE 332.

CE Major Status

Selected Elective

Course Objectives

In the current revolution of high-throughput experimental methods in genomics, biologists are relying more heavily than ever on computational analyses. In this capstone course, students explore software development for real problems that arise in the analysis of such data. Solving such problems often involves aspects of data structures, algorithm design and analysis, discrete mathematics, machine learning, statistics, molecular biology, and genetics. There is a real sense of exploration and discovery in this area.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context

- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

Each team designs, implements, and experiments with software for a current research problems in Computational Molecular Biology. The team tests its tool on real biological data and presents the results at the end of the quarter. The topic of the research project changes with each offering.

CSE 431 Introduction to Theory of Computation

Credits

3.0 (3 hrs. lecture)

Lead Instructor

Anup Rao

Textbook

- *Computational Complexity*, Arora

Course Description

Models of computation, computable and noncomputable functions, space and time complexity, tractable and intractable functions.

Prerequisites

either CSE 312 or CSE 322.

CE Major Status

Selected Elective

Course Objectives

Develop the concepts and skills necessary to be able to evaluate the computability and complexity of practical computational problems.

ABET Outcomes

(a) an ability to apply knowledge of mathematics, science, and engineering

Course Topics

- Turing machines (deterministic, nondeterministic, multitape)
- Church-Turing Thesis
- Decidability and undecidability, diagonalization, and reducibility
- Halting problem, Post correspondence problem, Rice's Theorem, and other undecidability results
- Time and space complexity
- P vs. NP, NP-completeness, Cook's Theorem, and other NP-complete problems
- PSPACE, PSPACE-completeness, PSPACE-complete problems
- L vs. NL, NL-completeness, Savitch's Theorem, Immerman-Szelepcsényi Theorem

CSE 440 Introduction to HCI: User Interface Design, Prototyping, and Evaluation

Credits

5.0 (3 hrs lecture, 2 hrs section)

Lead Instructor

James Landay

Textbook

None

Course Description

Human-Computer Interaction (HCI) theory and techniques. Methods for designing, prototyping, and evaluating user interfaces to computing applications. Human capabilities, interface technology, interface design methods, and interface evaluation tools and techniques.

Prerequisites

either CSE 326 or CSE 332.

CE Major Status

Selected Elective

Course Objectives

At the end of the course, students will understand the basics of user centered design. They will have completed a substantial group project, involving identifying, iteratively refining, prototyping, and testing a complex user interface. In addition, they will have written a final report on their project, made a video, presented the results in a talk, and made a poster.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning

(j) knowledge of contemporary issues

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Contextual inquiry and task analysis
- Sketching and storyboarding
- Paper prototyping
- Video prototyping
- Building web-based interactive prototypes
- History of human-computer interaction
- User testing
- Heuristic evaluation
- Interface metaphors

CSE 441 Advanced HCI: Advanced User Interface Design, Prototyping, and Evaluation

Credits

5.0 (5 hrs lecture/meeting times)

Lead Instructor

James Landay

Textbook

None

Course Description

Human-Computer Interaction (HCI) theory and techniques. Advanced methods for designing, prototyping, and evaluating user interfaces to computing applications. Novel interface technology, advanced interface design methods, and prototyping tools.

Prerequisites

CSE 440.

CE Major Status

None

Course Objectives

Gain a much deeper understanding of techniques in human computer interaction, including design methods, testing methods, and prototyping techniques, along with knowledge of when to best employ them. Work in teams to build and test a substantial HCI application.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Studio-based design and critiques
- Evaluation techniques for HCI (beyond those covered in CSE 440)
- Mobile user interface design
- Understanding and evaluating interfaces for behavior change
- Visual design

CSE 444 Database Systems Internals

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Magdalena Balazinska

Textbook

- *Database Systems*, Garcia-Molina

Course Description

The rational data model and the SQL query language. Conceptual modeling: entity/relationships, normal forms. XML, XPath, and XQuery. Transactions: recovery and concurrency control. Implementation of a database system. A medium sized project using a rational database backend.

Prerequisites

CSE 332; CSE 344.

CE Major Status

Selected Elective

Course Objectives

Databases are at the heart of modern commercial application development. Their use extends beyond this to many applications and environments where large amounts of data must be stored for efficient update and retrieval. The purpose of this course is to provide an introduction to the design and use of database systems, as well as an appreciation of the key issues in building such systems. We begin by covering the relational model and the SQL language. We then study methods for database design, covering the entity relationship model. Next, we discuss XML as a data model, and present languages for querying it. We see how XML is used for sharing data among different applications in a distributed environment. We then inspect the architecture of a database system, and discuss efficient storage of data, execution of queries and query optimization. Finally, we touch on some advanced topics in database systems.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

- (e) an ability to identify, formulate, and solve engineering problems
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- data models
- conceptual design
- query languages
- system components
- data storage
- query optimization
- transaction processing.

CSE 446 Machine Learning

Credits

3.0 (3 hrs lecture)

Lead Instructor

Dan Weld

Textbook

- *Machine Learning*, Murphy

Course Description

Methods for designing systems that learn from data and improve with experience. Supervised learning and predictive modeling: decision trees, rule induction, nearest neighbors, Bayesian methods, neural networks, support vector machines, and model ensembles. Unsupervised learning and clustering.

Prerequisites

either CSE 326 or CSE 332; either STAT 390, STAT 391, or CSE 312.

CE Major Status

Selected Elective

Course Objectives

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for

engineering practice

Course Topics

- Architecture of a Relation Extractor
- Supervised Learning & Logistic Regression
- Instaread and Features for ML
Project Discussion and Crawling the Web
- IR Models & Index Construction and Link Analysis & Pagerank
- SE Query Processing: Alta Vista
- NYU's 2011 KBP System
- Computational Advertising
- Crowdsourcing
- Cryptography & Practical Internet Security
- Mining unstructured healthcare data

CSE 451 Introduction to Operating Systems

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Edward Lazowska

Textbook

- *Operating Systems: Principles and Practice*, Thomas E. Anderson and Michael Dahlin

Course Description

Principles of operating systems. Process management, memory management, auxiliary storage management, resource allocation. No credit to students who have completed CSE 410 or E E 474.

Prerequisites

CSE 351 or CSE 378; CSE 326 or CSE 332; CSE 333..

CE Major Status

Selected Elective

Course Objectives

Give students a working knowledge of operating systems principles, design issues, algorithms and data structures. Build programming experience through a sequence of targeted OS projects.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve engineering problems
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- operating system structure, processes, threads, synchronization, scheduling, deadlock, virtual

- memory, secondary storage management, distributed systems, file systems, security

CSE 452 Introduction to Distributed Systems

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Thomas Anderson

Textbook

None

Course Description

Covers abstractions and implementation techniques in the construction of distributed systems, including cloud computing, distributed storage systems, and distributed caches.

Prerequisites

either CSE 444, CSE 451, or CSE 461.

CE Major Status

Selected Elective

Course Objectives

Give students a working knowledge of the principles, design issues, and algorithms underlying distributed systems. Build programming experience through a sequence of targeted distributed system projects.

ABET Outcomes

N/A

Course Topics

Client server computing, the web, cloud computing, peer-to-peer systems, and distributed storage systems.

Remote procedure call, preventing and finding errors in distributed programs, maintaining consistency of distributed state, fault tolerance, high availability, distributed lookup, and distributed security.

CSE 454 Advanced Internet and Web Services

Credits

5.0 (5 hrs lecture/meetings)

Lead Instructor

Daniel Weld

Textbook

None

Course Description

Design of Internet search engines, including spider architecture, inverted indices, frequency rankings, latent semantic indexing, hyperlink analysis, and refinement interfaces. Construction of scalable and secure web services. Datamining webserver logs to provide personalized and user-targeted services. Large project.

Prerequisites

CSE 326 or CSE 332.

CE Major Status

Selected Elective

Course Objectives

Understand the intellectual foundations of Web Search and Internet technologies including information retrieval, data mining, and cryptography. Be able to understand, build, and debug a Web Search Engine.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning

(j) knowledge of contemporary issues

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Internet history & future
- Architecture of an information extraction system
- Supervised learning, logistic regression & application to relation extraction
- Parsing, POS tags, named entity recognition & other features for extraction
- Crawling the Web
- Information retrieval models, index construction & compression
- Search engine query processing & Alta Vista case study
- Hyperlink analysis & fast external-memory Pagerank computation
- Case study: NYU's knowledge-base population system
- Mining unstructured healthcare data
- Computational advertising
- Crowdsourcing
- Cryptography & practical internet security

CSE 455 Computer Vision

Credits

4.0 (3 hrs lecture)

Lead Instructor

Steven Seitz

Textbook

None

Course Description

Introduction to image analysis and interpreting the 3D world from image data. Topics may include segmentation, motion estimation, image mosaics, 3D-shape reconstruction, object recognition, and image retrieval.

Prerequisites

CSE 303 or CSE 333; CSE 326 or CSE 332; recommended: MATH 308; STAT 391.

CE Major Status

Selected Elective

Course Objectives

Students learn the basics of computer vision and some of the state-of-the-art techniques. They will be able to write programs that can perform image segmentation, image matching, object detection or recognition, and applications such as content-based image retrieval or construction of panoramas. Upon completion of the course they should be able to take an internship or job with a vision company or research lab doing vision or to participate in undergraduate research leading to potential graduate level research.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve engineering problems
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- feature detection, descriptors, and matching
- image segmentation
- motion
- mosaics
- 3D sensing and reconstruction
- object recognition

CSE 456 Story Design for Computer Animation

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Barbara Mones

Textbook

None

Course Description

Animation principles and production for story development and design. Design, development, and production of several storyreels, which are a tool for the pre-production of animated features and shorts. Student use authoring tools to present finished work.

Prerequisites

None

CE Major Status

None

Course Objectives

To teach story structure for short animated film. To develop collaborative skills. To learn cinematography and the technology related to story-reel development. To learn about timing and improvisational acting and how this relates to short film [production for animated films. To understand technical troubleshooting for short animated films. To learn how to plan and create technical production notebooks for short animated films. To learn to use the technology to create story-reels for animated productions. To learn about new techniques for animation production.

ABET Outcomes

Course Topics

- Story Structure
- Story Structure for Animated Films
- Applying story structure techniques to analyze and improve the impact of animated shorts.
- Improv and Reference
- Physical Humor
- Cinematography

- Layout and Design
- Scoring an animated film
- Pitching stories
- Responding to feedback from a group.
- Planning for Motion
- Motion principles
- Storyboarding
- Character Design
- Story-reel Creation

CSE 457 Computer Graphics

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Brian Curless

Textbook

- *Interactive Computer Graphics*, Angel

Course Description

Introduction to computer image synthesis, modeling, and animation. Topics may include visual perception, displays and framebuffers, image processing, affine and projective transformations, hierarchical modeling, hidden surface elimination, shading, ray-tracing, anti-aliasing, texture mapping, curves, surfaces, particle systems, dynamics, character animation, and animation principles.

Prerequisites

CSE 303 or CSE 333; CSE 326 or CSE 332; recommended: MATH 308.

CE Major Status

Selected Elective

Course Objectives

Introduction to computer image synthesis and interactive computer graphics applications. Learn fundamentals of 2D and 3D computer graphics modeling, rendering, and animation through homework and projects.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for

engineering practice

Course Topics

- visual perception
- displays and framebuffers
- image processing
- affine and projective transformations
- hierarchical modeling
- hidden surface elimination
- shading
- ray-tracing
- anti-aliasing
- texture mapping
- curves
- surfaces
- particle systems
- dynamics
- character animation
- animation principles.

CSE 458 Computer Animation

Credits

5.0 (3 hrs lecture, 2 hours meetings)

Lead Instructor

Barbara Mones

Textbook

None

Course Description

Introduction to basic principles of computer generated animation. Focus on the modeling and lighting of animated characters. Students from art, CSE, and music team up on projects to be built on commercially-available modeling and lighting packages.

Prerequisites

either CSE 457, ART 380, or MUSIC 403.

CE Major Status

None

Course Objectives

To introduce students to the fundamentals of 3D modeling, shading, lighting, animating and rigging characters for three-dimensional computer generated environments. To understand the complex technical and aesthetic components of animation design.

ABET Outcomes

Course Topics

- Modeling digital objects that one can find reference for in the real world.
- Modeling hard surface and characters for 3D animated digital environments
- Shading objects.
- Lighting concepts from the real world applied to digital 3D environments
- Character Animation Principles
- Character Animation Projects
- Theory and fundamentals of character rigging for computer animation
- learning the basics of the animation pipeline for film production

CSE 459 Pre-Production for Collaborative Animation

Credits

5.0 (3 hrs lecture, 2 hours meetings)

Lead Instructor

Barbara Mones

Textbook

None

Course Description

Pre-production of collaboratively designed animated shorts. In-depth analysis of classical and computer generated works. Character design and pre-planning, model sheets, character rigging, storyreel and animatics, character motion, design for multiple characters, and principles of animation as applied to character motion and effects.

Prerequisites

CSE 458.

CE Major Status

None

Course Objectives

To be able to design and produce all of the materials needed for the pre-production of an animated short film. To be able to create concept art, reference material, animatics and signature shots that properly support a story. To work on a production team. To meet production deadlines and to troubleshoot technical and aesthetic problems.

ABET Outcomes

Course Topics

- story development for animated shorts
- small team production with an emphasis on the technical and aesthetic pipeline.
- new tools for character modeling
- setup, storyboard and layout,
- compositing for short films
- performance and animation
- animation planning
- character rigging

- signature shots
- signature shot production.

CSE 460 Animation Capstone

Credits

5.0 (3 hrs lecture, 2 hours meetings)

Lead Instructor

Barbara Mones

Textbook

None

Course Description

Apply the knowledge gained in previous animation courses to produce a short animated film. Topics include scene planning, digital cinematography, creature and hard surface modeling, animatics and basics of character animation, and rendering techniques.

Prerequisites

CSE 458, CSE 459.

CE Major Status

Selected Elective

Course Objectives

Students will work together to produce a short animated film using the story-reel, animatic, concept art and signature shots designed and produced in cse459. Students will work on several teams and take on leadership of one of the teams. The production will be required to spend considerable time applying all of the previous skills learned in 456, 458 and 459. Students will need to meet deadlines and work well in a group as every part of the production pipeline will need to succeed in order to create a fully completed film. Students will also create a poster and DVD to present the work that they've completed. Students will take part in screening the film to the campus community.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility

- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- advanced lighting
- advanced efx
- advanced animation
- advanced shading/texture
- advanced cinematography
- render quality
- meeting deadline
- renderfarm maintenance
- compositing and post production
- poster and DVD design

CSE 461 Introduction to Computer-Communication Networks

Credits

4.0 (3 hrs lecture, 1hr section)

Lead Instructor

Arvind Krishnamurthy

Textbook

- *Computer Networks*, Tannenbaum

Course Description

Computer network architectures, protocol layers, network programming. Transmission media, encoding systems, switching, multiple access arbitration. Network routing, congestion control, flow control. Transport protocols, real-time, multicast, network security.

Prerequisites

either CSE 326 or CSE 332; either CSE 303 or CSE 333.

CE Major Status

Selected Elective

Course Objectives

To provide students with an understanding of how to construct large-scale computer networks. This includes an appreciation of the fundamental problems that arise in building networks, the design principles that are of proven value, and the common implementation technologies that are in use today.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve engineering problems
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- This course introduces the basics of networking, ranging from transmitting bits over wires to the Web and distributed computing. We focus on the inter-networking issues in-between these two extremes. We will cover protocol layering in general and the following topics in particular:
 - framing
 - error correction
 - packet and circuit switching
 - multi-access protocols (Ethernet)
 - queuing
 - addressing and forwarding (IP)
 - distance vector and link state routing
 - reliable transport
 - congestion control (TCP)
 - security

CSE 464 Advanced Topics in Digital Animation

Credits

1.0 (seminar)

Lead Instructor

Barbara Mones

Textbook

None

Course Description

Students design individual animated works for professional quality demo reels. 2- and 3-D animatics, special effects design, advanced character animation techniques, 3-D paint techniques and integration, short design, sequence planning, non-photorealistic rendering options, interactive animation for pre-planning, and advanced production techniques and strategies.

Prerequisites

CSE 458.

CE Major Status

None

Course Objectives

Each student will produce and direct their own animated short project. This short may be used for their demoreel in order to apply for industry positions. Much of the work will be individual and some will be team based.

ABET Outcomes

N/A

Course Topics

- professional quality demo reels
- direction for animated shorts
- production roles for animated shorts
- advanced 2- and 3-D animatics
- advanced special effects design
- advanced character animation techniques
- 3-D paint techniques and integration
- shot and sequence planning
- options for non-photorealistic rendering
- animation for pre-visualization
- advanced production techniques and strategies

CSE 466 Software for Embedded Systems

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Joshua Smith

Textbook

MSP430 Microcontroller Basics, John H. Davies

Course Description

Software issues in the design of embedded systems. Microcontroller architectures and peripherals, embedded operating systems and device drivers, compilers and debuggers, timer and interrupt systems, interfacing of devices, communications and networking. Emphasis on practical application of development platforms.

Prerequisites

either CSE 352 or CSE 378; either CSE 303 or CSE 333.

CE Major Status

Selected Elective

Course Objectives

Understanding of basic microcontroller architecture and the motivation for their special features that distinguish them from microprocessors. Understanding of interfacing techniques for connecting microcontrollers to a variety of sensors and actuators (both digital and analog). An appreciation for the different considerations and constraints that software developers for embedded systems must deal with. Appreciation of power management methods. Understanding of basic communication protocols both wired and wireless. Facility with a complete set of tools for embedded systems programming and debugging. Experience with implementing several embedded systems with particular focus on their interaction between multiple devices and between devices and general-purpose computers.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues

Course Topics

- Embedded Systems: applications and platforms
- Microcontrollers: architectures, peripheral units, memory systems
- Interrupts: interrupt architectures, device drivers, timers
- Interfacing: basics of interfacing including GPIO, D/A, A/D, serial ports
- Communication Protocols: serial and parallel communication methods
- Introduction to Sensor Networks: basics of sensor networks including networking protocols
- Sensor Node Software: embedded and real-time OS issues, modularity
- Radio Protocols: common radio protocols and emerging standards
- Project Design: developing the specification of the final project (flock, soccer, etc.)
- TinyOS Case Study: bottom-up design of an embedded operating system
- Industry Guest Lectures: lectures from local companies developing mobile and embedded systems
- Special Topics: on-going embedded systems research projects at UW

CSE 467 Advanced Digital Design

Credits

4.0 (3 hrs lecture, 3 hr lab)

Lead Instructor

Bruce Hemingway

Textbook

- *Digital Design and Computer Architecture*, Harris

Course Description

Advanced techniques in the design of digital systems. Hardware description languages, combinational and sequential logic synthesis and optimization methods, partitioning, mapping to regular structures. Emphasis on reconfigurable logic as an implementation medium. Memory system design. Digital communication including serial/parallel and synchronous/asynchronous methods.

Prerequisites

either CSE 352 or CSE 370; either CSE 326 or CSE 332.

CE Major Status

Selected Elective

Course Objectives

1. To learn how to design digital systems, from specification and simulation to construction and debugging.
2. To learn techniques and tools for programmable logic design
3. To learn how to use modern laboratory test equipment, including logic analyzers and oscilloscopes.
4. To understand the limitations and difficulties in modern digital design, including wiring constraints, high-speed, etc.

5. To design, construct, test, and debug a moderate-scale digital circuit.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve engineering problems
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Overview of digital technology
- Logic families
- Electrical realities
- Computer-aided design
- Laboratory
- System-level components

CSE 471 Computer Design and Organization

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Susan Eggers

Textbook

- *Computer Architecture*, Hennessy

Course Description

CPU instruction addressing models, CPU structure and functions, computer arithmetic and logic unit, register transfer level design, hardware and micro-program control, memory hierarchy design and organization, I/O and system components interconnection. Laboratory project involves design and simulation of an instruction set processor.

Prerequisites

either CSE 352 or CSE 378.

CE Major Status

Selected Elective

Course Objectives

Teach the design and architecture of major components of the structure of the central processing unit and memory hierarchy of modern microprocessor systems. Use a cycle by cycle simulator to illustrate logic complexities.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data

- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (e) an ability to identify, formulate, and solve engineering problems
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Pipelining
- Branch prediction
- Exceptions
- Examples of CPU's in modern microprocessors
- Exploiting Instruction Level Parallelism
- Scoreboard and Tomasulo's algorithm
- Superscalars
- Caches and cache assists
- Hardware assists for paging systems
- TLB's
- Symmetric MultiProcessors
- Cache coherence
- Synchronization

CSE 472 Introduction to Computational Linguistics

Credits

5.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Glenn Slayden

Textbook

- *Speech & Language Processing*, Jurafsky

Course Description

Introduction to computer applications of linguistic theory, including syntactic processing, semantic and pragmatic interpretation, and natural language generation.

Prerequisites

either LING 200 or LING 400; either LING 461, CSE 311, or CSE 321.

CE Major Status

Selected Elective

Course Objectives

Be familiar with computational linguistic tools and resources, and how they are applied in research in both computational linguistics and other subfields.

Have a sense of the state of the art in this subfield.

Be able to conceptualize problems from the perspective of computational linguistics.

ABET Outcomes

n/a

Course Topics

- Finite state morphology
- Regular expressions
- Formal grammars; Chomsky hierarchy; Context-free grammars
- Bayes' theorem
- N-grams and Language Modeling
- Part-of-speech tagging
- Semantic representations
- Clustering and classifiers
- Evaluation: Precision and Recall
- Algorithms for corpus processing
- Feature-structures and unification-based grammars

CSE 473 Introduction to Artificial Intelligence

Credits

3.0 (3 hrs lecture)

Lead Instructor

Luke Zettlemoyer

Textbook

- *Artificial Intelligence*, Russell

Course Description

Principal ideas and developments in artificial intelligence: Problem solving and search, game playing, knowledge representation and reasoning, uncertainty, machine learning, natural language processing. Not open for credit to students who have completed CSE 415.

Prerequisites

CSE 326 or CSE 332; recommended: CSE 312; either STAT 390, or STAT 391.

CE Major Status

Selected Elective

Course Objectives

Mastery of the fundamental concepts and techniques of artificial intelligence.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and

safety, manufacturability, and sustainability

(e) an ability to identify, formulate, and solve engineering problems

(i) a recognition of the need for, and an ability to engage in life-long learning

(j) knowledge of contemporary issues

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Problem solving and search
- Knowledge representation and reasoning
- Reasoning under uncertainty
- Machine learning
- Planning

CSE 477 Digital System Design

Credits

5.0 (2 hrs lecture, 3 hrs lab, additional meetings times tbd)

Lead Instructor

Shwetak Patel

Textbook

None

Course Description

Capstone design experience. Prototype a substantial project mixing hardware, software, and communication components. Focuses on use of embedded processors and programmable logic in digital system design, case studies, and emerging components and platforms. Provides a complete experience in embedded system design and management.

Prerequisites

CSE 451; CSE 466; CSE 467.

CE Major Status

Selected Elective

Course Objectives

To serve as a capstone design course to tie together the computer engineering curriculum via the design of a complete embedded system involving multiple communicating components. To gain appreciation for the interaction between hardware and software in embedded system design. Familiarity with basic inter-component communication methods. To experience the development of a complete product from design to implementation and debugging. To present design goals and decisions as well as implementation results in both verbal presentation and written documentation. To work toward a common goal in a team environment. Students will prototype a substantial

project that mixes hardware, software, and communication components. Lectures will be limited and will be focused on topics relating to advance prototyping, emerging components and platforms, case studies, and other topics of interest to the class.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Introduction: Embedded Systems; Product design; Product development process
- Team-based Design: Project definition; Decomposition into hardware and software components;
- Evaluation of needs; Assessment of team members skills; Experimental design to resolve unknowns;
- Design reuse
- Project Evolution: Design for maintainability; Design for upgradability; Modularity in Design
- Research Directions: Architecture innovations; Operating systems; Networking; Application domains;
- User interfaces
- Case Studies: Past capstone design projects; Industry case studies; Research projects in department
- Advanced prototyping
- Product requirement documents

CSE 481A Capstone Software Design: Operating Systems

Credits

5.0 (3 hrs lecture, 2 hrs+ meeting times)

Lead Instructor

Gary Kimura

Textbook

None

Course Description

Students work in teams to design and implement a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product.

Prerequisites

CSE 331 or CSE 341; CSE 326 or CSE 332; CSE 351 or CSE 378; substantial programming experience such as CSE 451 or CSE 457.

CE Major Status

Selected Elective

Course Objectives

Build on the fundamental computer operating system knowledge learned in the introductory course and see how those concepts are actually incorporated in the Windows operating system. Student teams work on adding extensions to the operating system and gain valuable software engineering experience.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- The Windows Operating System internals provides a fundamental basis for the lectures, but this material is highly augmented with the engineering fundamentals that went into designing and building the system.

CSE 481B Capstone Software Design: Tablet PC

Credits

5.0 (3 hrs lecture, 2 hrs+ meeting times)

Lead Instructor

Richard Anderson

Textbook

None

Course Description

Students work in teams to design and implement a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product.

Prerequisites

CSE 331 or CSE 341; CSE 326 or CSE 332; CSE 351 or CSE 378; substantial programming experience such as CSE 451 or CSE 457.

CE Major Status

Selected Elective

Course Objectives

To learn about the software design process through hands-on development of a software product. To experience working in larger teams than you have had to deal with previously in our curriculum. To experience building sophisticated applications by making use of real-world tools, rather than trying to build everything from scratch. To gain experience dealing with the usability issues related to mobile devices. To have some fun (by building a cool application). To develop a portfolio documenting your efforts that could be useful in looking for a job. To gain experience in demonstrating and promoting a prototype application.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Software development in teams; basics of pen based computation. Most student time is spent in the development process, and performing critiques of it.

CSE 481C Capstone Software Design: Robotics

Credits

5.0 (3 hrs lecture, 2 hrs+ meeting times)

Lead Instructor

Rajesh Rao

Textbook

None

Course Description

Students work in teams to design and implement a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product.

Prerequisites

CSE 331 or CSE 341; CSE 326 or CSE 332; CSE 351 or CSE 378; substantial programming experience such as CSE 451 or CSE 457.

CE Major Status

Selected Elective

Course Objectives

This course will teach students to understand the key concepts underlying autonomous systems interacting with the real world. By implementing and applying different approaches, the students will learn how to model and control real world systems using probabilistic methods. The programming component of this course will enable the students to solve large scale, open-ended problems in a team setting.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- robot control
- probabilistic sensor models
- Bayesian state estimation (Kalman and particle filters)
- robot localization and mapping
- path planning
- multi-robot coordination.

CSE 481D Capstone Software Design: Games

Credits

5.0 (3 hrs lecture, 2 hrs+ meeting times)

Lead Instructor

John Zahorjan

Textbook

None

Course Description

Students work in teams to design and implement a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product.

Prerequisites

CSE 331 or CSE 341; CSE 326 or CSE 332; CSE 351 or CSE 378; substantial programming experience such as CSE 451 or CSE 457.

CE Major Status

Selected Elective

Course Objectives

To learn about the software design process through hands-on development of a software product. To experience working in larger teams than you have had to deal with previously in our curriculum. To experience building sophisticated applications by making use of real-world tools, rather than trying to build everything from scratch. To gain experience dealing with the performance demands of high-performance, real-time, distributed applications. To have some fun (by building a game). To develop a portfolio documenting your efforts that could be useful in looking for a job.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- software development in teams
- basics of games design and development
- most student time is spent in the development process, and performing critiques of it.

CSE 481H Capstone Software Design: Accessibility

Credits

5.0 (3 hrs lecture, 2 hrs+ meeting times)

Lead Instructor

Alan Borning

Textbook

None

Course Description

Students work in teams to design and implement a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product.

Prerequisites

CSE 331 or CSE 341; CSE 326 or CSE 332; CSE 351 or CSE 378; substantial programming experience such as CSE 451 or CSE 457.

CE Major Status

Selected Elective

Course Objectives

Smart phones and other mobile devices are revolutionizing the tools available to people with disabilities. The objectives of this capstone are to design, build, and test accessibility applications on mobile devices, working with mentors who are either themselves disabled, or who work closely with the blind, deaf, or other communities. In the process, students will learn about working on teams and the software design process, and also about the needs and concerns of people with disabilities with respect to information technology. They will also further develop presentation and writing skills in the context of the project report, project video, and final poster session.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- software development in teams
- programming on the Android or iPhone mobile platforms
- developing project proposals and mockups
- building a prototype implementation on an Android or iPhone device
- techniques for testing mobile applications with people with disabilities
- producing a short video, writing a final report, presenting the work at a poster session

CSE 481I Capstone Software Design: Sound

Credits

5.0 (3 hrs lecture, 2 hrs+ meeting times)

Lead Instructor

Bruce Hemingway

Textbook

None

Course Description

Students work in teams to design and implement a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product.

Prerequisites

CSE 331 or CSE 341; CSE 326 or CSE 332; CSE 351 or CSE 378; substantial programming experience such as CSE 451 or CSE 457.

CE Major Status

Selected Elective

Course Objectives

Students will work in teams to design, implement, and release a software project utilizing some of the techniques listed in Course topics.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within

realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(d) an ability to function on multi-disciplinary teams

(e) an ability to identify, formulate, and solve engineering problems

(f) an understanding of professional and ethical responsibility

(g) an ability to communicate effectively

(h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context

(i) a recognition of the need for, and an ability to engage in life-long learning

(j) knowledge of contemporary issues

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

The course participants will design and build projects utilizing computer audio techniques for human interfacing, sound recording and playback, encoding and decoding, synchronization, sound synthesis, recognition, and/or analysis/resynthesis.

CSE 481K Capstone Software Design: Resource-constrained Environments

Credits

5.0 (3 hrs lecture, 2 hrs+ meeting times)

Lead Instructor

Ruth Anderson

Textbook

None

Course Description

Students work in teams to design and implement a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product.

Prerequisites

CSE 331 or CSE 341; CSE 326 or CSE 332; CSE 351 or CSE 378; substantial programming experience such as CSE 451 or CSE 457.

CE Major Status

Selected Elective

Course Objectives

Students form interdisciplinary project groups to scope and design projects for resource-constrained environments. The emphasis is on group work leading to the creation of testable realizations and completion of initial evaluations of the software and hardware artifacts produced. Students work in inter-disciplinary groups with a faculty or graduate student manager. Groups document their work in the form of posters, verbal presentations, videos, and written reports.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Software development in teams
- Project platforms vary but often include a mobile component
- Attention to design for conditions of low connectivity, low cost, different user populations
- Most student time is spent in the development process, and performing critiques of it.

CSE 4810 Capstone Software Design: 3D Cameras

Credits

5.0 (3 hrs lecture, 2 hrs+ meeting times)

Lead Instructor

Dieter Fox

Textbook

None

Course Description

Students work in teams to design and implement a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product.

Prerequisites

CSE 331 or CSE 341; CSE 326 or CSE 332; CSE 351 or CSE 378; substantial programming experience such as CSE 451 or CSE 457.

CE Major Status

Selected Elective

Course Objectives

- To gain appreciation for the challenges in developing complex sensor driven computing systems.
- To experience the development of a complete sensor-driven system from design to implementation.
- To present design goals and decisions as well as implementation results in both verbal presentation and written documentation.
- To have you work in a larger team than in the past to learn about coordinating such groups.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Course Topics

- Kinect style depth cameras: functionality and SDKs
- Object recognition
- Body pose tracking
- Human computer interaction
- World Wide Telescope

CSE 484 Computer Security

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Tadayoshi Kohno

Textbook

- *Foundations of Security*, Daswani

Course Description

Foundations of modern computer security, including software security, operating system security, network security, applied cryptography, human factors, authentication, anonymity, and web security.

Prerequisites

either CSE 326 or CSE 332; either CSE 351 or CSE 378.

CE Major Status

Selected Elective

Course Objectives

Give students a foundation in modern computer security. A primary goal is to help enable students to think about computer security (i.e., develop the security mindset) so that they can apply that mindset when evaluating new products and new technologies in the future. As part of this mindset, students are taught to consider the ethical issues surrounding computer security (e.g., decisions to / not to patch vulnerabilities, decisions to / not to disclose vulnerabilities, implications with releasing new technologies with unexpected (by consumers) security and privacy properties). Additional goals are to make students conversant in modern cryptography, software security, Web security, HCI-security, network security, mobile device security, and malware.

ABET Outcomes

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi-disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues

Course Topics

- Ethics
- Threat modeling and risk management
- Software security
- Cryptography
- Web security
- HCI-security
- Network security
- Mobile device security
- Malware
- Advanced topics

CSE 486 Introduction to Synthetic Biology

Credits

3.0 (3 hrs lecture)

Lead Instructor

Georg Seelig

Textbook

U. Alon, Control Systems Engineering, An Introduction to Systems Biology: Design Principles of Biological Circuits, Chapman and Hall, 2006.

Course Description

Mathematical modeling of transcription, translation, regulation and metabolism in cell; computer aided design methods for synthetic biology; implementation of information processing, Boolean logic and feedback control laws with genetic regulatory networks; modularity, impedance matching and isolation in biochemical circuits; parameter estimation methods.

Prerequisites

either MATH 136 or MATH 307, AMATH 351, or CSE 321 and MATH 308 or AMATH 352.

CE Major Status

Selected Elective

Course Objectives

At the end of this course students will be able to:

1. Understand the challenges and applications of synthetic biology.
2. Understand the basic cellular processes including transcription, translation, regulation, metabolism, and information processing.
3. Build mathematical models of biochemical systems inside cells using Boolean logic, finite state machines, ordinary differential equations and/or stochastic processes.
4. Understand biochemical processes in terms of stability, robustness, parameter sensitivity, modularity, and evolvability.
5. Estimate model parameters from data.
6. Use Matlab or similar software to model, design and simulate systems.

7. Use molecular sensors, regulatory elements, reporters, enzymes, etc. in new designs and predict their behavior mathematically.
8. Understand the risks and ethical considerations of synthetic biology.

ABET Outcomes

- (a) An ability to apply knowledge of mathematics, science, and engineering. Lectures and homework deal with the application of differential equations, linear algebra and Laplace transforms to control systems.
- (c) An ability to design a system, component or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability and sustainability. Students are required to apply the skills acquired in this course to design control systems to meet specific performance requirements.
- (e) An ability to identify, formulate and solve engineering problems. Some of the homework assignments require students to evaluate different design approaches to reach an acceptable design.

Course Topics

1. The applications of synthetic biology
2. The risks, ethics and challenges of synthetic biology
3. Transcription, translation and regulation
4. Metabolism
5. Review of mathematical modeling
6. Mass action and enzyme kinetics
7. Stochastic chemical kinetics
8. Modeling software
9. In vitro synthetic biology
10. Composition, modularity and sensitivity
11. Robustness and sensitivity in biochemical systems
12. Parameter estimation and system identification
13. Review of recent literature in synthetic biology

CSE 487 Advanced Systems and Synthetic Biology

Credits

3.0

Lead Instructor

Georg Seelig

Textbook

U. Alon, Control Systems Engineering, An Introduction to Systems Biology: Design Principles of Biological Circuits, Chapman and Hall, 2006.

Course Description

This course assumes a basic understanding of synthetic and systems biology. It introduces a variety of advanced and more in-depth topics on cellular networks, including their operation and engineering. The course is intended for engineering and computer science students. Topics include advanced mathematical modeling of cellular networks; computational standards in systems and synthetic biology; computer algorithms for computational analysis; metabolic flux analysis, control and engineering; protein signaling pathways, analysis, control and engineering.

Prerequisites

either BIOEN 401, BIOEN 423, E E 423, or CSE 486

CE Major Status

Selected Elective

Course Objectives

At the end of this course students will be able to:

- Understand the different modeling approaches used to represent cellular networks (Structural, Continuous and Stochastic Approaches)
- Understand the differences between the fundamental cellular subsystems, metabolic, protein and genetic and how this influences potential engineering approaches.
- Develop an appreciation for the need for standards and ontologies in model exchange and part representation.
- Understand, implement and use a variety of computational approaches including FBA, MFA, Bifurcation and evolutionary methods.

- Understand the basic principles of metabolic control including small signal analysis and elementary mode analysis.
- Learn how to carry out a robustness analysis of a metabolic pathway and propose strategies for engineering pathways.
- Understand the control of protein networks, highlighting differences and similarities with genetic and metabolic systems.
- Use computational analysis to study the dynamic properties of protein networks and the design of robust systems.

ABET Outcomes

- (a) An ability to apply knowledge of mathematics, science, and engineering. Lectures and homework deal with the application of differential equations, linear algebra and Laplace transforms to control systems.
- (c) An ability to communicate effectively.

Course Topics

1. The importance of network structure in cellular networks
2. Review of continuous and stochastic models of cellular networks
3. The interplay between structure and dynamics
4. Bifurcation analysis and evolutionary design approaches in synthetic biology.
5. Standards and ontologies (SBML, CellML, PoBoL, CAD in synthetic biology)
6. Control systems in metabolism
7. Control systems in protein networks
8. Robustness and small signal analysis of cellular pathways
9. Advanced structural analysis including elementary modes, FBA and MFA
10. Metabolic engineering strategies
11. Protein networks, control and dynamical analysis
12. Protein network engineering

CSE 488 Laboratory Methods in Synthetic Biology

Credits

4.0 (1 hr lecture, 3 hr lab)

Lead Instructor

Georg Seelig

Textbook

Nelson and Cox, Principles of Biochemistry, 4th Edition, Freeman, 2004.

Micklos, Freyer and Crothy, DNA Science Cold Spring Harbor Laboratory Press, 2003.

Course Description

Designs and builds transgenic bacterial using promoters and genes taken from a variety of organisms. Uses construction techniques including recombination, gene synthesis, and gene extraction. Evaluates designs using sequencing, fluorescence assays, enzyme activity assays, and single cell studies using time-lapse microscopy.

Prerequisites

either BIOEN 401, BIOEN 423, E E 423, or CSE 486

CE Major Status

Selected Elective

Course Objectives

At the end of this course students will be able to:

1. Culture bacteria.
2. Manipulate DNA with restriction, ligation, PCR and gel electrophoresis.
3. Transform bacteria with recombinant DNA and screen for successful transformants.
4. Design genetic regulatory networks at the level of the DNA sequence.
5. Extract DNA from cells and prepare it for sequencing.
6. Perform fluorescence and growth assays with a fluorescence plate reader.
7. Use a fluorescence microscope to capture single cell behavior in time.
8. Analyze experimental data and fit it to mathematical models.
9. Understand the risks and ethical considerations of synthetic biology.

ABET Outcomes

- (a) An ability to apply knowledge of mathematics, science, and engineering to the design of biochemical networks for specific applications.
- (b) An ability to design and conduct experiments, as well as to analyze and interpret data
- (c) An ability to design a system, component or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability and sustainability.
- (d) An ability to function on multi-disciplinary teams.
- (f) An understanding of professional and ethical responsibilities related to introducing new genetic material into the ecosystem.
- (h) The broad education necessary to understand the impact of engineering biological solutions in a global, economic, environmental and societal context.
- (j) Knowledge of contemporary issues in genetic engineering, gene therapy, biofuels and energy, medicine and disease.
- (k) An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Course Topics

1. The applications, risks and ethics of synthetic biology
2. Lab safety
3. Basic lab techniques including pipetting and sterile technique
4. Bacterial cultures and growth curves
5. Design of experiments and controls
6. Extraction of plasmid DNA from *E. coli*
7. Recombinant DNA techniques include restriction digests, gel purification, ligation, and PCR based methods
8. Sequencing for the purposes of debugging constructs
9. Fluorescence reporters and methods for measuring cell activity using fluorescence
10. Time lapse fluorescence microscopy
11. The application of differential equations and stochastic processes to predicting the behavior of synthetic biochemical networks
12. Parameter estimation and system identification

CSE 490 Patent Law for Engineers

Credits

2.0 (2 hrs lecture)

Lead Instructor

Ben Dugan

Textbook

None

Course Description

Lectures, discussions, and possibly labs on topics of current interest in computer science and engineering not covered by other CSE undergraduate courses.

Prerequisites

None

CE Major Status

Selected Elective

Course Objectives

Perplexed, annoyed, or interested in patents? Confused by copyright laws? This course provides a survey of intellectual property law for a technical (non-legal) audience. The purpose of the course is to assist engineers and scientists in navigating and utilizing various intellectual property regimes effectively in the business context. In the patent space, we will study the significant revisions of U.S. patent law under the America Invents Act of 2011, including the change from a first-to-invent to a first-to-file patent system and new post-grant review procedures. Additional patent-related topics will include patent preparation and prosecution, claim interpretation, and assessing patent validity and infringement. Other intellectual property areas will also be addressed, including copyright, trademark, and trade secret law. The course will balance the discussion of practical legal considerations with broader policy questions (e.g., should certain

subject matter be off limits for patenting? the relationship between innovation and intellectual property regimes, etc.).

ABET Outcomes

- (f) an understanding of professional and ethical responsibility
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) knowledge of contemporary issues

Course Topics

- Week 1: [Survey of IP Law](#)
- Week 2: [Reading a Patent](#)
- Week 3: [Patent Prosecution, Provisional Applications, Priority, Foreign Rights](#)
- Week 4: [Conditions for Patentability](#)
- Week 5: [Claim Drafting](#)
- Week 6: [Patent Analysis](#)
- Week 7: [Copyright & Open Source Licenses](#)
- Week 8: [Software Patents](#)
- Week 9: [Patent Reform](#)
- Week 10: [Wrap-up](#)

CSE 498 Senior Project

Credits

1.0 – 6.0 (variable, independent research)

Lead Instructor

Not applicable

Textbook

None

Course Description

A report (and perhaps demonstration) describing a development, survey, or small research project in computer science or an application to another field. Objectives: (1) integrating material from several courses, (2) introducing the professional literature, (3) gaining experience in writing a technical document, and (4) showing evidence of independent work. Work normally extends over more than one quarter, for a maximum of 6 credits for CSE 498; 9 credits are required for CSE 498H.

Prerequisites

None

CE Major Status

None

Course Objectives

n/a

ABET Outcomes

n/a

Course Topics

Not applicable

Appendix B – Faculty Vitae

Richard Anderson

1. Education – PhD, Computer Science, Stanford University, 1986.

2. Academic experience

1998-present Professor, Department of Computer Science and Engineering, University of Washington, Seattle, WA

2003-2009 Associate Chair for Educational Programs, Department of Computer Science and Engineering, University of Washington, Seattle, WA

1991-1998 Associate Professor, Department of Computer Science and Engineering, University of Washington, Seattle, WA

1993-1994 Visiting Professor of Computer Science, Indian Institute of Science, Bangalore, India

1986-1991 Assistant Professor, Department of Computer Science and Engineering, University of Washington, Seattle, WA

1985-1986 Postdoctoral Research Fellow, Mathematical Sciences Research Institute, Berkeley, CA

3. Non-academic experience

2001-2002 Visiting Researcher, Learning Sciences and Technology Group, Microsoft Research

2009-present Visiting Computer Scientist, PATH, Seattle, WA

4. Certifications or professional registrations: None

5. Current membership in professional organizations: ACM

6. Honors and awards:

1987 National Science Foundation Presidential Young Investigator Award

1993 Indo-American Fellowship, Indo-US Subcommission on Education and Culture

2007 Faculty Innovator for Teaching Award, College of Engineering, UW

2008 Premier Award for Excellence in Engineering Education Courseware

7. Service activities:

Dev 2013 Conference Papers co-chair

College of Engineering, Council on Promotion and Tenure, 2011-2104

8. Selected recent publications:

1. Fiore-Silfast, B, Iyengar K, Iyengar S, **Anderson R**, et al. Mobile video for patient education: The midwives' perspective, Association for Computing Machinery **3rd Annual Symposium on Computing for Development**, 2013.
2. **Anderson R**, Roberston, C., Nabi, E., Sahni, S., and Setia, T., Facilitated Video Instruction in Low Resource Schools, Information and Communication Technologies and Development 2012.
3. **Anderson R**, Lloyd J, and Newland S, Software for National Level Vaccine Cold Chain Management, Information and Communication Technologies and Development 2012.
4. Chaudhri R, Borriello G, **Anderson R**, et al. FoneAstra: Enabling remote monitoring of vaccine cold chains using commodity mobile phones. Association for Computing Machinery **1st Annual Symposium on Computing for Development**, 2010.
5. Saxena A, Sahni U, Gupta R, Arora A, **Anderson R**, et al. Evaluating Facilitated Video Instruction for Primary Schools in Rural India, Information and Communication Technologies and Development 2010.
6. **Anderson R**, Blantz E, Lubinski D, et al. Smart Connect: Data connectivity for peripheral health facilities. *NSDR 2010: Proceedings of the 4th ACM Workshop on Networked Systems for Developing Regions*. 2010.
7. Linnell N, **Anderson RJ**, Prey J. Cross-Cultural Issues in a Tutored Video Instruction Course, **Special Interest Group on Computer Science Education** Technical Symposium on Computer Science Education, 2008.
8. **Anderson RJ**, Anderson R, Davis P, et al. Classroom presenter: Enhancing interactive education with digital ink. *IEEE Computer Society*. September 2007;40(9):56–61.
9. Briefly list the most recent professional development activities: None

Ruth E. Anderson

1. Education
 - Ph. D., Computer Science & Engineering, University of Washington, 2006
 - M.S., Computer Science & Engineering, University of Washington, 1994
 - B.S., Mathematical Sciences: Computer Science, University of North Carolina, 1991
2. Academic experience
 - Teaching Faculty, Lecturer, University of Virginia, Dept. of Computer Science, Charlottesville, VA. (Fall 2000 - Spring 2005).
3. Non-academic experience
 - Software Developer, IBM Research, T. J. Watson Research Center, Hawthorne, NY. (Summer 1991),
 - Software Developer, IBM Corporation, Research Triangle Park, NC. (Summers 1989 & 1990)
4. Certifications or professional registrations
5. Current membership in professional organizations
 - ACM, ACM-SIGCSE, ACM-SIGCHI, ASEE
6. Honors and awards
 - ACM Faculty Award, voted best teacher by students, UVa Department of Computer Science (2004)
 - Bob Bandes Memorial Award for excellence in teaching, UW Dept. of Computer Science & Eng. (1992)
7. Service activities (within and outside of the institution)
 - Associate Program Chair, ACM Special Interest Group on Computer Science Education Symposium 2013 (SIGCSE 2013)
 - Workshops Co-Chair, ACM Special Interest Group on Computer Science Education Symposium 2010 & 2011 (SIGCSE 2010 & SIGCSE 2011)
 - NSF Review Panels: CE21, TUES
8. Publications
 - Designing Technology for Resource-Constrained Environments: Three Approaches to a Multidisciplinary Capstone Sequence, Ruth E. Anderson, Beth Kolko, Gaetano Borriello, Richard J. Anderson. Proceedings of 38th ASEE/IEEE Frontiers in Education Conference (FIE 2012), Seattle, WA, USA. October 2012.
 - The Midwife's Assistant: Designing Integrated Learning Tools to Scaffold Ultrasound Practice, Alexis Hope, Waylon Brunette, Wayne Gerard, Jacqueline Keh,

- Laura Schlenke, Ruth Anderson, Michael Kawooya, Rob Nathan, Beth Kolko. The 5th International ACM Conference on Information & Communication Technologies and Development (ICTD 2012), Atlanta, GA, USA. March 2012.
- Introducing Parallelism and Concurrency in the Data Structures Course, Dan Grossman, Ruth E. Anderson. SIGCSE 2012: Technical Symposium on Computer Science Education, Raleigh, NC, USA. February 2012.
 - Reducing Maternal Mortality: An Ultrasound System for Village Midwives, Waylon Brunette, Matthew Hicks, Alexis Hope, Ginger Ruddy, Ruth E. Anderson, Beth Kolko. The 1st IEEE Global Humanitarian Technology Conference 2011(GHTC 2011), Seattle, WA, USA. November 2011.
 - Deciding to Major in Computer Science: A Grounded Theory of Students' Self-Assessment of Ability, Colleen M. Lewis, Ken Yasuhara, Ruth E. Anderson. International Computing Education Research Workshop (ICER'11), Providence, RI, USA, August 2011.
 - Designing Technology for Resource-Constrained Environments: A Multidisciplinary Capstone Sequence, Ruth E. Anderson, Beth Kolko. 118th ASEE Annual Conference & Exposition, Vancouver, BC, Canada, June 2011.
 - Portable Antenatal Ultrasound Platform for Village Midwives, Waylon Brunette, Wayne Gerard, Matthew A. Hicks, Alexis Hope, Mitchell Ishimitsu, Pratik Prasad, Ruth E. Anderson, Gaetano Borriello, Beth E. Kolko, Robert Nathan. The 1st Annual Symposium on Computing for Development (ACM DEV 2010), London, UK. December 2010.
 - Experiences with a Transportation Information System that Uses Only GPS and SMS, Ruth E. Anderson, Waylon Brunette, Erica Johnson, Caitlin Lustig, Anthony Poon, Cynthia Putnam, Odina Salihbaeva, Beth E. Kolko, Gaetano Borriello. The 4th International ACM/IEEE Conference on Information & Communication Technologies and Development (ICTD 2010), London, UK. December 2010.
 - An Approach to Integrating ICTD Projects into an Undergraduate Curriculum, Ruth E. Anderson, Richard Anderson, Gaetano Borriello, and Joyojeet Pal. SIGCSE 2010: Technical Symposium on Computer Science Education, Milwaukee, WI, USA. March 2010.
 - Capstone Projects as Community Connectors, Ruth E. Anderson, Gaetano Borriello, Hélène Martin, and Leonard Black. Consortium for Computing Sciences in Colleges-Northwestern Conference (CCSC-NW), Tacoma, WA, USA. October 2009.
 - Building a Transportation Information System Using Only GPS and Basic SMS Infrastructure, Ruth E. Anderson, Anthony Poon, Caitlin Lustig, Waylon Brunette, Gaetano Borriello, and Beth E. Kolko. The 3rd International ACM/IEEE Conference on Information & Communication Technologies and Development (ICTD 2009), Doha, Qatar. April 2009.
 - Collaborative Technologies in International Distance Education, Richard Anderson, Ruth Anderson, Natalie Linnell, Mansoor Pervaiz, Umar Saif, and Fred Videon. The 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2009), Santiago, Chile. April 2009.

Thomas E. Anderson

1. Education
A.B., Philosophy, Harvard University, 1983
M.S., Computer Science, University of Washington, 1989
PhD, Computer Science, University of Washington, 1991
2. Academic experience
University of Washington, Robert E. Dinning Professor, 2009-present, full-time
ETH Zurich, Visiting Professor, 2009, full-time
University of Washington, Professor, 2001-09, full-time
University of Washington, Associate Professor, 1997-2001, full-time
University of California-Berkeley, Associate Professor, 1996-97, full-time
University of California-Berkeley, Assistant Professor, 1991-96, full-time
3. Non-academic experience
Asta Networks, Co-founder and interim CEO/CTO, 2000-01
Digital Equipment Corporation, Research Intern, 1990
4. Certifications or professional registrations: None
5. Current membership in professional organizations: ACM, Fellow
IEEE, Koji Kobayashi Award for Computing and Communications
PlanetLab Consortium, Steering Committee Chair
6. Honors and awards
ACM Fellow; IEEE Koji Kobayashi Award for Computing and Communications; ACM SIGOPS Mark Weiser Award; Diane S. McEntyre Award for Excellent in Teaching; NSF Presidential Faculty Fellowship; Alfred P. Sloan Research Fellowship; NSF Young Investigator Award; IBM Graduate Fellowship
7. Service activities
Program Committee Chair, ACM Symposium on Operating Systems Principles, 2009
Program Committee Co-Chair, ACM SIGCOMM'06 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, 2006
Program Committee Co-Chair, Third Workshop on Hot Topics in Networks (HotNets-III), 2004
Program Chair, USENIX Symposium on Internet Technologies and Systems, 2001
Program Committee Member, ACM SIGCOMM 2000, 2001, 2002, 2004
8. Selected Publications
“F10: Fault Tolerant Engineered Networks”, Vincent Liu, Dan Halperin, Arvind Krishnamurthy, and Thomas Anderson. *NSDI* 2013.

“[Operating Systems: Principles and Practice](#)”, Thomas Anderson and Michael Dahlin, Recursive Books, 2012.

“[FreeDOM: a New Baseline for the Web](#)”, Raymond Cheng, Will Scott, Arvind Krishnamurthy, and Thomas Anderson. *HotNets* 2012.

“[LIFEGUARD: Practical Repair of Persistent Route Failures](#)”, Ethan Katz-Bassett, Colin Scott, David Choffnes, Italo Cunha, Vytautas Valancius, Nick Feamster, Harsha Madhyastha, Thomas Anderson, and Arvind Krishnamurthy. *SIGCOMM* 2012.

“An architecture for zero-configuration P2P web applications [Scalable Consistency in Scatter](#)”, Lisa Glendenning, Ivan Beschastnikh, Arvind Krishnamurthy, and Thomas Anderson, *SOSP* 2011.

“[ETTM: A Scalable Fault Tolerant Network Manager](#)”, Colin Dixon, Hardeep Uppal, Vjekoslav Brajkovic, Dane Brandon, Thomas Anderson, and Arvind Krishnamurthy, *NSDI* 2011.

“[Privacy-preserving P2P Data Sharing with OneSwarm](#)”, Tomas Isdal, Michael Piatek, Arvind Krishnamurthy, and Thomas Anderson, *SIGCOMM* 2010.

“[Resolving IP Aliases with Prespecified Timestamps](#)”, Justine Sherry, Ethan Katz-Bassett, Mary Pimenova, Harsha Madhyastha, Arvind Krishnamurthy, And Thomas Anderson. *IMC* 2010.

“[Reverse Traceroute](#)”, Ethan Katz-Bassett, Harsha Madhyastha, Vijay Adhikari, Colin Scott, Justine Sherry, Peter van Wesep, Thomas Anderson and Arvind Krishnamurthy. Best Paper, *NSDI* 2010.

9. Briefly list the most recent professional development activities
Co-Founder and Steering Committee Chair, PlanetLab Consortium, 2003 – present
Co-Founder and Steering Committee Co-Chair, ACM/USENIX Network Systems Design and Implementation Conference, 2002 – present
Scriptroute, a open programmable Internet monitoring system. www.scriptroute.org
BitTyrant, a high-performance, strategic BitTorrent client. bittyrant.cs.washington.edu
OneSwarm, privacy-preserving peer-to-peer file sharing. oneswarm.cs.washington.edu
iPlane Nano, real-time predictions of Internet performance embedded in peer-to-peer applications. iplane.cs.washington.edu
Reverse Traceroute, real time diagnosis of the reverse Internet path from destination to source. revtr.cs.washington.edu
Hubble, a real-time monitor for Internet black holes. hubble.cs.washington.edu

Magdalena Balazinska

1. Education

PhD in Computer Science, MIT, Cambridge, MA, 2006
M.S in Electrical Eng., Ecole Polytechnique de Montreal, 2000
B.S in Computer Eng., Ecole Polytechnique de Montreal, 2000

2. Academic experience

University of Washington, Computer Science & Engineering
Associate Professor, Sep. 2012 – Present

University of Washington, Computer Science & Engineering
Assistant Professor, Jan. 2006 – Sep. 2012

3. Non-academic experience

IBM Thomas J. Watson Research Center, Hawthorne, NY, Intern, May – Aug 2002
Motorola Canada Software Center, Montreal, Quebec, Software Engineer, Jan. – Jul. 2000

4. Certifications or professional registrations: None

5. Current membership in professional organizations : IEEE and ACM SIGMOD

6. Honors and awards

Best lightning talk at the 6th XLDB conference 2012
2nd place in the Best Paper Award Competition for the Challenges and Visions track at VLDB 2011
Best student paper award, 2011, The 5th International Open Cirrus Summit
10-year most influential paper award, 2010, WCRE
HaLoop paper invited to the Best of VLDB 2010 issue of VLDB Journal, VLDB 2010
NSF CAREER Award, 2009, NSF
HP Labs Innovation Research Award, 2009 and 2010, HP Labs
Microsoft Research New Faculty Fellow, 2007
Rogel Faculty Support Award, 2006
Microsoft Research Graduate Fellowship, 2003-2005
Best student paper award, 2002, USENIX Security Symposium
NSERC scholarship for graduate studies, 2000-2002

7. Service activities (within and outside of the institution)

Service within the institution (some examples only):

UW Certificate in Database Management advisory board member. UW's Professional & Continuing Education (November 2012 – Present).

Guest speaker at the high school programming competition, Fall 2012. Title “Big Data Management: (Some) Challenges and Opportunities”.

Member of the CSE executive committee (September 2008 -June 2010).

Member of the undergraduate admissions committee (Fall 2009, 2011-2012, and 2012-2013).

Service outside the institution (some examples only)

SSDBM 2013 PC chair.

SIGMOD 2012 demonstration chair.

Program committee area chair: ICDE 2011, ICDE 2013.

Regular PC member for SIGMOD, VLDB, ICDE, and CIDR

8. Briefly list the most important publications and presentations from the past five years – title, co-authors if any, where published and/or presented, date of publication or presentation

SkewTune: Mitigating Skew in MapReduce Applications

YongChul Kwon, Magdalena Balazinska, Bill Howe, and Jerome Rolia

SIGMOD 2012. **Most cited paper from that conference as of Spring 2013 with 37 citations.**

Data Markets in the Cloud: An Opportunity for the Database Community

Magdalena Balazinska, Bill Howe, and Dan Suciu

PVLDB, Vol 4, Nb 12, 2011 (VLDB 2011)

2nd place in the Best Paper Award Competition for the Challenges and Visions track

Yingyi Bu, Bill Howe, Magdalena Balazinska, and Michael D. Ernst. HaLoop: Efficient iterative data processing on large clusters. Proc. of the VLDB Endowment (PVLDB), 3(1):285–296, 2010. **Most cited paper from that conference as of Spring 2013.**

Selected for Best of VLDB 2010 issue of VLDB Journal. Over 180 citations.

Building the Internet of things using RFID: the RFID Ecosystem experience

E Welbourne, L Battle, G Cole, K Gould, K Rector, S Raymer, M Balazinska, and

Gaetano Borriello. Internet Computing, IEEE 13 (3), 48-55. 2009. **Nearly 200 citations.**

9. Briefly list the most recent professional development activities

Regular attendance at various conferences and workshops in data management.

Paul Beame

1. Education –
PhD Computer Science, University of Toronto, 1987
M.Sc. Computer Science, University of Toronto, 1982
B.Sc. Mathematics, University of Toronto, 1981
2. Academic experience –
Professor, Computer Science & Engineering, U. Washington, 1999–Present.
Member, Institute for Advanced Study, Princeton, 2010–11
Associate Professor, Computer Science & Engineering, U. Washington, 1993–99.
Visiting Associate Professor, Computer Science, U. Toronto, 1993–94.
Assistant Professor, Computer Science & Engineering, U. Washington, 1987–93.
Postdoctoral Associate, Laboratory for Computer Science, Massachusetts Institute of Technology, 1986–87
Lecturer, part time, Computer Science & Engineering, U. Toronto, Summer 1985.
3. Non-academic experience -
Consultant, Microsoft Research 2001-02
4. Certifications or professional registrations
5. Current membership: ACM, IEEE Computer Society
6. Honors and awards:
Honorable Mention 2008 IJCAI-JAIR Best Paper Prize (sole runner-up)
ACM Recognition of Service Award 2006
Presidential Young Investigator, National Science Foundation
7. Selected service activities:
National and international:
Chair, ACM SIGACT, 2012- Present,
Chair, IEEE Computer Society TCMF, 2008-11
General Chair, IEEE FOCS Conference 2009-11
Vice-chair IEEE Computer Society TCMF 2002-07
Steering Committee, IEEE Symposium on Logic in Computer Science 2008-13 Steering Committee, IEEE Conference on Computational Complexity 2007-10 Executive Committee ACM SIGACT 2001-05
University of Washington Service:
Associate Chair for Education, CSE, 2009-2010, 2011-Present

CSE Space Committee (Chair) 2002-10, 2011-Present
CSE Faculty Program Advisor BS/MS Program 2008-10

8. Selected Publications:

Paul Beame, Paraschos Koutris, and Dan Suciu. Communication Steps for Parallel Query Processing. In *Proceedings of the 32nd Annual Symposium on Principles of Database Systems*, New York, NY, June 2013. To appear.

Paul Beame, Chris Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, pages 212–232, New York, NY, May 2012. Also, invited submission to special issue of *SIAM J. Computing* for STOC 2012.

Paul Beame and Widad Machmouchi. The quantum query complexity of AC^0 . *Quantum Information & Computation*, 12(7–8):670–676, 2012.

Paul Beame and Dang-Trinh Huynh-Ngoc (Trinh Huynh). Multiparty communication complexity and threshold circuit size of AC^0 . In *Proceedings 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 53–62, Atlanta, GA, October 2009. Also, *SIAM Journal on Computing*, 41(3):484–518, 2012.

Paul Beame and Dang-Trinh Huynh-Ngoc (Trinh Huynh). On the value of multiple read/write streams for approximating frequency moments. In *Proceedings 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 499–508, Philadelphia, PA, October 2008. Also, *ACM Transactions on Computation Theory*, 3(2):6.1–6.22, 2011.

Paul Beame, Russell Impagliazzo, Toniann Pitassi, and Nathan Segerlind. Formula caching in DPLL. *ACM Transactions on Computation Theory*, 1(3):9:1–33, March 2010.

Gaetano Borriello

1. Education –
Ph.D., 1988, Computer Science, University of California, Berkeley, CA
M.S., 1981, Electrical Engineering, Stanford University, Palo Alto, CA
B.S., 1979, Electrical Engineering, Polytechnic University, Brooklyn, NY
2. Academic experience –
Professor, Computer Science & Engineering, U. Washington, 1998–Present
Associate Professor, Computer Science & Engineering, U. Washington, 1993–98
Assistant Professor, Computer Science & Engineering, U. Washington, 1988–93
Visiting Professor, Scuola Superiore Sant’Anna, Pisa, Italy, 1995-6
Visiting Scientist, Inter-University Microelectronics Center, Leuven, Belgium, 1992
3. Non-academic experience –
Visiting Scientist, Google Inc., Seattle, WA, 2008-09
Intel Corporation, Founding Director, Intel Research Seattle, 2001-03
Member of Research Staff, Xerox Palo Alto Research Center, Palo Alto, CA, 1980-87
4. Certifications or professional registrations
5. Current membership: ACM, IEEE Computer Society
6. Honors and awards –
Jerre D. Noe Endowed of Professorship, University of Washington, 2006-present
Honorable Mention, University of Washington, Graduate Mentor Award, 2012 and 2013
University of California, Berkeley CS Division Distinguished Alumni, 2012
Fellow of the Institute of Electrical and Electronics Engineers, 2011
Fellow of the Association for Computing Machinery, 2010
Divisional Recognition Award – Corporate Technology Group, Intel Corporation, 2003
Fulbright Pisa Chair Award – Scuola Superiore Sant’Anna, Pisa, Italia, 1995
Distinguished Teaching Award – University of Washington, 1995
Jr Faculty Achievement Award – Coll. of Engineering, University of Washington, 1994
Presidential Young Investigator Award – National Science Foundation, 1988
7. Selected service activities –
DARPA Information Science and Technology, 2010-13
ACM Symposium on Computing for Development, Steering Committee, 2010-present
IEEE Pervasive Computing Magazine, Advisory Board, 2008-present
NSF/CISE Advisory Board, 2001-03
Associate Chair for Education, UW/CSE, 1998-2000, 2010-11
Associate Chair for Research, UW/CSE, 2006-08
Undergraduate Program Coordinator, UW/CSE, 1997-2001, 2012-13

8. Selected Publications –

H. Hile, R. Grzeszczuk, A. Liu, R. Vedantham, J. Kosecka, G. Borriello. Landmark-Based Pedestrian Navigation with Enhanced Spatial Reasoning, 7th International Conference on Pervasive Computing, Nara, Japan, May 2009.

A. Liu, H. Hile, G. Borriello, P. Brown, M. Harniss, H. Kautz, K. Johnson. Customizing Directions in an Automated Wayfinding System for Individuals with Cognitive Impairment, 11th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS09), Pittsburgh, PA, October 2009.

Y. Anokwa, C. Hartung, W. Brunette, A. Lerer, C. Tseng, G. Borriello. Open Data Kit: Tools to Build Information Services for Developing Regions, 4th IEEE/ACM International Conference on Information and Communication Technologies and Development, London, England, December 2010.

B. DeRenzi, G. Borriello, J. Jackson, V. Kumar, T. Parikh, P. Virk, N. Lesh. Mobile Phone Tools for Field-Based Health Care Workers in Low-Income Countries. *Mount Sinai Journal of Medicine: A Journal of Translational and Personalized Medicine*, Vol. 78, No. 3, pp. 406–418, May/June 2011.

N. Dell, N. Breit, T. Chaluco, J. Crawford, G. Borriello. Digitizing Paper Forms with Mobile Imaging Technologies, ACM 2nd Annual Symposium on Computing for Development (DEV), Atlanta, Georgia, March 2012.

Y. Anokwa, N. Ribeka, T. Parikh, G. Borriello, M. Were. Design of a Phone-Based Clinical Decision Support System for Resource-Limited Settings, 5th IEEE/ACM International Conference on Information and Communication Technologies and Development, Atlanta, Georgia, March 2012.

W. Brunette, R. Chaudhri, R. Sodt, M. Goel, M. Falcone, J. Van Orden, G. Borriello. Open Data Kit Sensors: A Framework for Application-Level Sensor Drivers on Android Devices. 10th International Conference on Mobile Systems, Applications, and Services (Mobisys12), Lake District, United Kingdom, June 2012.

R. Chaudhri, G. Borriello, R. Anderson. Monitoring Vaccine Cold Chains in Developing Countries, *IEEE Pervasive Computing* (special issue on Technologies for Development – ICT4D), Vol. 11, No. 3, July-September 2012.

E. Larson, M. Goel, S. Heltshe, M. Rosenfeld, G. Borriello, S. Patel. SpiroSmart: Using a Microphone to Measure Lung Function on a Mobile Phone. 14th International Conference on Ubiquitous Computing (UbiComp 2012), Pittsburgh, PA, September 2012. (Best paper nominee.)

9. Briefly list the most recent professional development activities

Luis Ceze

1. Education

University of Illinois at Urbana Champaign (UIUC), Urbana, IL. Ph.D. in Computer Science, July 2007. Thesis: *Bulk Operation and Data Coloring for Multiprocessor Programmability*

University of São Paulo, Polytechnic School, Brazil. M.Eng. in Electrical Engineering, August 2002. Thesis: *BGLsim, Complete System Simulator for Blue Gene/L*

University of São Paulo, Polytechnic School, Brazil. B.S. in Electrical Engineering, December 2000

2. Academic experience

Computer Science and Engineering, University of Washington Seattle, WA.
Associate Professor, September 2012-present.
Assistant Professor, September 2007-August 2012.

3. Non-academic experience

IBM Research, Yorktown Heights, NY. Research Intern/Co-op, June 2001-July 2002, Summers of 2003, 2004, 2006.

Consultant for MSR and Coresync (UW startup for which I was a co-founder)

4. Certifications or professional registrations

5. Current membership in professional organizations: ACM, IEEE and Usenix.

6. Honors and awards

2013 Paper selected for the IEEE Micro Top Picks, Jan/Feb 2013

2010 Kavli Frontiers of Science Fellow

2010 UW CSE ACM Undergraduate Teacher of the Year Award

2010 Sloan Research Fellowship

2009 Paper selected for the IEEE Micro Top Picks, Jan/Feb 2010

2009 Microsoft New Faculty Fellowship

2009 NSF CAREER Award

2008 Paper on thesis work selected to appear in Communication of ACM

2008 Two papers selected for the IEEE Micro Top Picks, Jan/Feb 2009

2008 Paper selected from ISCA'08 to appear in Communications of ACM

2008 David J. Kuck Outstanding Thesis Award, UIUC

2005 Paper selected for the IEEE Micro Top Picks, Jan/Feb 2006

2005 WJ Poppelbaum Memorial Award, UIUC
2004 IBM Outstanding Internship Award
2004-2005 IBM PhD Fellowship

7. Service activities

Served on UW-CSE executive committee, 2011-2013.

Invited instructor at ACACES Summer School, July 2013.

Invited participant of DARPA ISAT Workshop on Resilience, Mar 2013.

USENIX HotPar 2013, PC Member.

ACM Student Research Competition 2013, Committee Member.

Member of the USENIX HotPar Steering Committee, 2012-.

Member of Editorial Board for ACM Transactions on Parallel Computing.

Invited participant of NSF ACAR (Advancing Computer Architecture Research), and DARPA ISAT on Advancing Computer Systems.

IEEE Micro Top Picks 2013, Program Committee Member.

8. Selected Publications

Richard Muscat, Karin Strauss, Luis Ceze, and Georg Seelig. DNA-based Molecular Architecture with Spatially Localized Components. In *International Symposium on Computer Architecture (ISCA)*, June 2013

Brandon Lucia and Luis Ceze. Cooperative Empirical Failure Avoidance for Multithreaded Programs. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, March 2013

Hadi Esmaeilzadeh, Adrian Sampson, Luis Ceze, and Doug Burger. Neural Acceleration for General- Purpose Approximate Programs. In *International Symposium on Microarchitecture (MICRO)*, December 2012. Selected for IEEE Micro Top Picks 2012

9. Briefly list the most recent professional development activities

Brian L. Curless

1. Educational history

Stanford University, Ph.D. in Electrical Engineering, June 1997

Stanford University, M.S. in Electrical Engineering, June 1991

University of Texas at Austin, B.S. in Electrical Engineering, May 1988

2. Academic experience

Professor, University of Washington, Department of Computer Science & Engineering (2010 – present).

Associate Professor, University of Washington, Department of Computer Science & Engineering (2003 – 2010).

Assistant Professor, University of Washington, Department of Computer Science & Engineering (1998 – 2003).

3. Non-academic experience

Scientific Advisory Board, Paraform, Inc., Mountain View, CA (1998 – 2002).

Software Contractor, Silicon Graphics Inc., Mountain View, CA (1993).

Research Engineer, Remote Measurements Laboratory at SRI International, Menlo Park, CA (1988 - 1989).

4. Certifications or professional registrations: n/a

5. Professional organizations

Association for Computing Machinery (1998-present)

IEEE Computer Society (1998-present)

6. Honors and awards

UW ACM Teaching Award, University of Washington, (2004)

Sloan Fellowship for Computer Science, University of Washington (2000)

NSF CAREER Award, University of Washington (1999)

Stanford Computer Science Department Arthur Samuel Thesis Award (1997)

Achievement Rewards for College Scientists (ARCS) fellowship (1993)

Gores Award for Teaching Excellence, Stanford (1992)

Solid State Industrial Affiliates Fellowship, Stanford (1990)

Graduated Summa cum Laude, University of Texas (1988)

7. Selected Service Activities

Co-editor-in-chief, Foundations and Trends in Computer Graphics and Computer Vision (2003 – present).

General Chair, 3D Vision 2013, Seattle, WA, June 29-30.

Program committee member for ACM SIGGRAPH, 2003, 2004, 2006, 2007, 2009, 2010, and 2013.

Professional Masters Program Faculty Coordinator, UW CSE (2005-2008, 2012-).

Faculty Recruiting Committee, UW CSE (2008-10, 2010-2011 (chair), 2012-2013).

Executive Committee, UW CSE (2003-2004, 2010-2011).

Chair Search Committee, UW CSE (2000-2001).

8. Selected Recent Publications

Shan Qi, Brian Curless, Yutaka Furukawa, Steven M. Seitz. The Visual Turing Test for 3D Scene Reconstruction. *3DV (3D Vision) 2013*, June 2013.

Robert T. Held, Ankit Gupta, Brian Curless, Maneesh Agrawala. 3D Puppetry: A Kinect-based Interface for 3D Animation. *ACM UIST 2012*.

Qi Shan, Sameer Agarwal, Brian Curless. Refractive Height Fields from Single and Multiple Images. *IEEE CVPR 2012*.

Juliet Fiss, Aseem Agarwala, Brian Curless. Candid Portrait Selection from Video. *SIGGRAPH Asia / ACM Transactions on Graphics*, December 2011.

Pravin Bhat, Brian Curless, Michael F. Cohen, and C. Larry Zitnick. GradientShop: An Optimization Framework for Image and Video Processing, *ACM Transactions on Graphics*, 2010.

Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *IEEE CVPR 2010*.

Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Reconstructing building interiors from images. In *IEEE ICCV 2009*.

9. Recent Professional Development Activity

Sabbatical Visitor, University of California, Berkeley, Department of Computer Science, (2011-2012).

Pedro Domingos

1. Education
1997: Ph.D., Information & Computer Science, University of California, Irvine.
1994: M.S., Information & Computer Science, University of California, Irvine.
1992: M.S., Electrical & Comp. Eng., Instituto Superior Tecnico, Lisbon, Portugal.
1988: B.S., Electrical & Comp. Eng, Instituto Superior Tecnico, Lisbon, Portugal.
2. Academic experience
2012-present: Professor, Computer Science & Eng., U. Washington.
2004-12: Associate Professor, Computer Science & Engineering, U. Washington.
1999-2004: Assistant Professor, Computer Science & Engineering, U. Washington.
1997-99: Assistant Professor, Instituto Superior Tecnico, Lisbon, Portugal.
3. Non-academic experience
4. Certifications or professional registrations
5. Current membership in professional organizations
ACM, ACM SIGKDD, AAI, IMLS.
6. Honors and awards
Fellowships: AAI, Fulbright, Sloan, Kavli, IBM, NATO, UC Regents.
Best paper awards: NIPS-12, UAI-11, EMNLP-09, PKDD-05, KDD-99, KDD-98.
NSF CAREER Award.
7. Service activities
Director of Professional Master's Program (2009-12).
Program co-chair: SRL-09, KDD-03.
Editorial board member: Machine Learning, JAIR, Applied Intelligence, EIS, IDA.
Co-founder and board member, International Machine Learning Society.
Area chair or program committee member for over 50 conferences and workshops, including AAI, CogSci, ICML, IJCAI, KDD, NIPS, SIGMOD, SRL, UAI, WWW.
Reviewer: Machine Learning, JAIR, JMLR, DMKD, IDA, Applied Intelligence, IEEE Computer, etc.
8. Briefly list the most important publications and presentations from the past five years – title, co-authors if any, where published and/or presented, date of publication or presentation

P. Domingos, "A Few Useful Things to Know about Machine Learning,"
Communications of the ACM, vol.55, no.10, 2012.

R. Gens and P. Domingos, “Discriminative Learning of Sum-Product Networks,” in *Advances in Neural Information Processing Systems 25* (pp. 3248-3256), Lake Tahoe, CA, 2012.

P. Domingos and W. Webb, “A Tractable First-Order Probabilistic Logic,” in *Proc. Twenty-Sixth AAAI Conference on Artificial Intelligence* (pp. 1902-1909), Toronto, Canada, 2012.

H. Poon and P. Domingos, “Sum-Product Networks: A New Deep Architecture,” in *Proc. Twenty-Seventh Conference on Uncertainty in Artificial Intelligence* (pp. 337-346), Barcelona, Spain, 2011.

V. Gogate and P. Domingos, “Probabilistic Theorem Proving,” in *Proc. Twenty-Seventh Conference on Uncertainty in Artificial Intelligence* (pp. 256-265), Barcelona, Spain, 2011.

V. Gogate and P. Domingos, “Approximation by Quantization,” in *Proc. Twenty-Seventh Conference on Uncertainty in Artificial Intelligence* (pp. 247-255), Barcelona, Spain, 2011.

C. Kiddon and P. Domingos, “Coarse-to-Fine Inference and Learning for First-Order Probabilistic Models,” in *Proc. Twenty-Fifth AAAI Conference on Artificial Intelligence* (pp. 1049-1056), San Francisco, CA, 2011.

H. Poon and P. Domingos, “Unsupervised Semantic Parsing,” in *Proc. 2009 Conference on Empirical Methods in Natural Language Processing* (pp. 1-10), Singapore, 2009.

P. Singla and P. Domingos, “Lifted First-Order Belief Propagation,” in *Proc. Twenty-Third AAAI Conference on Artificial Intelligence* (pp. 1094-1099), Chicago, IL, 2008.

9. Briefly list the most recent professional development activities

Oren Etzioni

1. Education
Carnegie Mellon University, School of Computer Science, Pittsburgh, PA.
Ph.D. January 1991; M.S., Computer Science, August 1988.
Harvard University, Cambridge, MA.
Honors B.A. in Computer Science, June 1986.
2. Academic experience
2005- present - Professor, Department of Computer Science and Engineering,
University of Washington
1996-2005 - Associate Professor, Department of Computer Science and Engineering,
University of Washington.
1991-1996 - Assistant Professor, Department of Computer Science and Engineering,
University of Washington
3. Non-academic experience
2010-present Co-Founder /CTO, Decide.com
2004-2008 Founder, Farecast, Inc (acquired by Microsoft)
2000- present Venture Partner, Madrona Venture Group
1999-2000 CTO, Go2net, Inc
1996-1997 Co-Founder/Chief Scientists, Netbot, Inc.
4. Certifications or professional registrations
4 US patents:
Co-Creator, the MetaCrawler service at www.metacrawler.com
Co-Creator, the BRUTE data mining software
Co-Creator, the LCW closed-world reasoning software package
5. Current membership in professional organizations: President, AI Access Foundation.
6. Honors and awards
2009 Washington Research Foundation Endowed Entrepreneurship Professorship in
Computer Science and Engineering
2007 Robert S. Engelmores Memorial Lecture Award
2005 Distinguished Paper Award, IJCAI '05 for *A Probabilistic Model of Redundancy
in Information Extraction.*
2003 AAAI Fellow
1993 NSF Young Investigator Award
1992 NSF Research Initiation Award
1987-2000 AT&T Bell Laboratories Fellow
7. Service activities

AAAI **Executive Council; Advisory Board** for *Journal of Artificial Intelligence Research*; **Editorial Boards** for *Computational Intelligence*, *Journal of Artificial Intelligence Research* (term ended, 2000), *Journal of Data Mining and Knowledge Discovery*; AAAI **Senior PC Member& PC member**, WWW, **Vice Chair** for Search (1999), Third International Conference on Autonomous Agents, 1999 (Program Chair). First International Conference on Autonomous Agents, 1997 (Area Chair for Software Agents); **Senior member** , International Joint Conference on Artificial Intelligence, 1993; **Reviewer**: NSF, NASA, *Machine Learning*, *Artificial Intelligence*, *Journal of Artificial Intelligence Research*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

8. Selected Publications

- [1] Stefan Schoenmackers, Jesse Davis, Oren Etzioni, and Daniel S. Weld, “**Learning First-Order Horn Clauses from Web Text,**” in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, 2010
- [2] Thomas Lin, Mausam, and Oren Etzioni, “**Identifying Functional Relations in Web Text,**” in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, 2010
- [3] Doug Downey, Oren Etzioni, and Stephen Soderland, “Analysis of a Probabilistic Model of Redundancy in Unsupervised Information Extraction,” *Artificial Intelligence*, 174(11):726-748, 2010.
- [4] Alan Ritter, Mausam, and Oren Etzioni, “A Latent Dirichlet Allocation method for Selectional Preferences,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp.424-434), Uppsala, Sweden, 2010.
- [5] Thomas Lin, Oren Etzioni and James Fogarty, “Identifying Interesting Assertions from the Web,” in *Proceeding of the 18th ACM conference on Information and knowledge management*, Hong Kong, China, 2010.
- [6] Stephen Soderland, Christopher Lim, Mausam, Bo Qin, Oren Etzioni, and Jonathan Pool, “Lemmatic Machine Translation,” in *Proceedings of the 12th Machine Translation Summit* (pp.128-135), Ottawa, Canada, 2009
- [7] Mausam, Stephen Soderland, Oren Etzioni, Daniel S. Weld, Michael Skinner, and Jeff Bilmes, “Compiling a Massive, Multilingual Dictionary via Probabilistic Inference,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (pp.262-270), Suntec, Singapore, 2009.
- [8] Stefan Schoenmackers, Oren Etzioni, and Daniel S. Weld, “Scaling Textual Inference to the Web,” in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (pp. 79-88), Honolulu, HI, 2008

9. Briefly list the most recent professional development activities: None

Ali Farhadi

1. Education
PhD. Computer Science, University of Illinois at Urbana-Champaign, 2011
2. Academic experience
Assistant Professor, Computer Science and Engineering, University of Washington, 2012-present
Postdoctoral Fellow, Robotics Institute, Carnegie Mellon University, 2011
Research Assistant, University of Illinois, 2005-2011
3. Non-academic experience
Adobe Research, Research intern, Summer 2009
Co-founder, Snap and Buy 2007
4. Certifications or professional registrations
5. Current membership in professional organizations
6. Honors and awards
CVPR2011 Best student Paper award
Google Inaugural Fellowship in Computer vision and image interpretation
Illinois Venture
Cognitive Science and Artificial Intelligence Award, Beckman Institute
Outstanding Research Award, UIUC
7. Service activities
Program committee member at CVPR 2007,2008,2009,2010,2011,2012,2013
Program committee member at ICCV 2009,2011,2013
Program committee member at ECCV 2008,2010, 2012
Reviewer for IJCV, TPAMI, PR
8. Selected Publications
Describing Objects by their Attributes
Ali Farhadi, Ian Endres, Derek Hoiem, David Forsyth
In proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), 2009.
Attribute-Centric Recognition for Cross-Category Generalization
Ali Farhadi, Ian Endres, Derek Hoiem
In proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), 2010.

Learning to Recognize Activities from a Wrong Viewpoint

Ali Farhadi and Mostafa Kamali

In proceedings of European conference on Computer Vision (ECCV'08), 2008.

Every Picture Tells a Story: Generating Sentences for Images

Ali Farhadi, Mohsen Hejrati, Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, David Forsyth

In proceedings of European conference on Computer Vision (ECCV'10), 2010.

Recognition Using Visual Phrases

Ali Farhadi, Amin Sadeghi

In proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), 2011.[Best Student Paper Award]

A Latent Model of Discriminative Aspect,

Ali Farhadi, Mostafa Kamali, Ian Endres, David Forsyth

In proceedings of International Conference on Computer Vision (ICCV'09), 2009.

Transfer Learning in Sign Language

Ali Farhadi, David Forsyth, and Ryan White

In proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07), 2007.

Attribute Discovery via Predictable Discriminative Binary Codes

Mohammad Rastegari, Ali Farhadi, David Forsyth

in Proceedings of European Conference on Computer Vision (ECCV'12), 2012.

Multi-Attribute Queries: To Merge or Not to Merge?

Mohammad Rastegari, Ali Diba, Devi Parikh, Ali Farhadi

in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13), 2013.

Adding Unlabeled Samples to Categories by Learned Attributes

Jonghyun Choi, Mohammad Rastegari, Ali Farhadi, Larry Davis

in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13), 2013.

9. Professional Development Activities

James Fogarty

1. Education:
Ph.D. Human Computer Interaction, Carnegie Mellon University, 2006
B.S. Computer Science, Virginia Tech, 2000
2. Academic Experience:
Associate Professor, Computer Science & Engineering, U. Washington, 2011-Present
Assistant Professor, Computer Science & Engineering, U. Washington, 2006-2011
3. Non-Academic Experience:
Consultant, Microsoft Research, 2007, 2010
4. Certifications or Professional Registrations:
5. Current Membership:
Association for the Advancement of Artificial Intelligence (AAAI)
Association for Computing Machinery (ACM)
Institute of Electrical and Electronics Engineers (IEEE)
6. Honors and Awards:
National Science Foundation CAREER Award, 2010
Best Paper, CHI 2010 (top 1% of 1346 submissions)
Best Paper Nominee, UbiComp 2009 (top 3 of 251 submissions)
Best Paper Nominee, CHI 2009 (top 5% of 1130 submissions)
Best Paper, CHI 2005 (top 1% of 372 submissions)
7. Selected Service Activities:
National and International:
Chair, CHI 2012 Subcommittee on Technology, Systems, and Tools
Program Committee, CHI 2008, 2009, 2010, 2012, 2013
Program Committee, UIST 2006, 2008, 2010
Governing Board Member, Human Computer Interaction Consortium, 2008-present
Chair, IJCAI 2009 Workshop on Intelligence and Interaction

University of Washington Service:
DUB Interdisciplinary Committee, 2009/10-present
UW CSE Graduate Admissions and Recruiting HCI Area Organizer, 2009/10-2012/13
UW CSE Undergraduate Admissions Committee, 2010/11
8. Selected Publications:

Epstein, D.A., Borning, A., and Fogarty, J. (2013). Fine-Grained Sharing of Sensed

Physical Activity: A Value Sensitive Approach. International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013), to Appear.

Dixon, M., Fogarty, J., and Wobbrock, J.O. (2012). A General-Purpose Target-Aware Pointing Enhancement Using Pixel-Level Analysis of Graphical Interfaces. ACM Conference on Human Factors in Computing Systems (CHI 2012), pp. 3167-3176.

Amershi, S., Fogarty, J., and Weld, D.S. (2012). ReGroup: Interactive Machine Learning for On-Demand Group Creation in Social Networks. ACM Conference on Human Factors in Computing Systems (CHI 2012), pp. 21-30.

Patel, K., Drucker, S.M., Fogarty, J., Kapoor, A., Tan, D.S. (2011). Using Multiple Models to Understand Data. International Joint Conference on Artificial Intelligence (IJCAI 2011), pp. 1723-1728.

Dixon, M., Leventhal, D., and Fogarty, J. (2011). Content and Hierarchy in Pixel-Based Methods for Reverse Engineering Interface Structure. ACM Conference on Human Factors in Computing Systems (CHI 2011), pp. 969-978.

Patel, K., Bancroft, N., Drucker, S., Fogarty, J., Ko, A., Landay, J.A. Gestalt: Integrated Support for Implementation and Analysis in Machine Learning Processes. (2010). ACM Symposium on User Interface Software and Technology (UIST 2010), pp. 37-46.

Dixon, M. and Fogarty, J. (2010). Prefab: Implementing Advanced Behaviors Using Pixel-Based Reverse Engineering of Interface Structure. ACM Conference on Human Factors in Computing Systems (CHI 2010), pp. 1525-1534.

Amershi, S., Fogarty, J., Kapoor, A., and Tan, D. (2010). Examining Multiple Potential Models in End User Interactive Concept Learning. ACM Conference on Human Factors in Computing Systems (CHI 2010), pp. 1357-1360.

Amershi, S., Fogarty, J., Kapoor, A., and Tan, D. Overview-Based Example Selection in End-User Interactive Concept Learning. (2009). ACM Symposium on User Interface Software and Technology (UIST 2009), pp. 247-256.

Froehlich, J., Larson, E., Campbell, T., Haggerty, C., Fogarty, J., and Patel, S.N. (2009). HydroSense: Infrastructure-Mediated Single-Point Sensing of Whole-Home Water Activity. International Conference on Ubiquitous Computing (UbiComp 2009), pp. 235-244.

9. Professional Development Activities: none

Dieter Fox

1. Education

- Dr. rer.-nat. (Ph.D.), University of Bonn, Germany
Computer Science: December 1998.
- Diplom (M.Sc.), University of Bonn, Germany
Computer science (major) and physics (minor): April 1993.
- Vordiplom (B.Sc.), University of Bonn, Germany
Computer science (major), economics and physics (minor): March 1990.

2. Academic experience

- Associate Professor, University of Washington, Department of Computer Science & Engineering: 2005 – present.
- Adjunct Associate Professor, University of Washington, Department of Electrical Engineering: 2005 – present.
- Assistant Professor University of Washington, Department of Computer Science & Engineering: 2000 – 2005.
- Adjunct Assistant Professor University of Washington, Department of Electrical Engineering: 2000 – 2005.
- Postdoctoral research associate, Carnegie Mellon University, School of Computer Science: 1998 – 2000.

3. Non-academic experience

- Director, Intel Labs Seattle, Seattle, WA: 2009 – 2011.

4. Certifications or professional registrations

5. Current membership in professional organizations

- IEEE, ACM, AAI.

6. Honors and awards

- IEEE Senior Member, 2012.
- AIJ Prominent Paper Award: Artificial Intelligence Journal, 2012.
- Fellow of the AAI: Association for the Advancement of Artificial Intelligence, 2011.
- AAI Outstanding Paper Award [1 out of 453 papers]: Conference on Artificial Intelligence, 2004.
- NSF CAREER Award, 2001.
- ICRA Best Paper Award [1 out of 1,100 papers]: IEEE International Conference on Robotics & Automation, 2000.

- AAAI Outstanding Paper Award [3 out of 475 papers]: Conference on Artificial Intelligence, 1998.
- IROS Best Paper Award [1 out of 476 papers]: IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998.

7. Service activities

- UW CSE 5th Year Masters Program (chair), 2011 - present.
- UW CSE graduate admissions committee (member), 2003.
- Program Chair: Robotics: Science and Systems (RSS), 2013.
- Editor: IEEE Transactions on Robotics (T-RO), since 2010.
- Conference Committee Chair Association for the Advancement of Artificial Intelligence (AAAI), 2009 – 2012.
- Advisory Board: Journal of Artificial Intelligence Research (JAIR), since 2011.
- Associate Editor: Journal of Artificial Intelligence Research (JAIR), 2008 – 2011.
- RSS Foundation Board, since 2009.
- Program co-chair: Conference on Artificial Intelligence (AAAI), 2008.

8. Selected Publications

- P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research (IJRR)*, 31(4), 2012.
- M. Krainin, P. Henry, X. Ren, and D. Fox. Manipulator and object tracking for in hand 3d object modeling. *International Journal of Robotics Research (IJRR)*, 30(9), 2011.
- J. Ko and D. Fox. GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *Autonomous Robots*, 27(1), 2009.
- L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for RGB-D based object recognition. In *Proc. of the International Symposium on Experimental Robotics (ISER)*, 2012.
- C. Matuszek, N. Fitzgerald, L. Zettlemoyer, and D. Fox. A joint model of language and perception for grounded attribute learning. In *Proc. of the International Conference on Machine Learning (ICML)*, 2012.

9. Professional Development Activities

Shyamnath Gollakota

1. Education
Ph.D., Electrical Engineering and Computer Science, MIT, 2013
2. Academic experience
University of Washington, Assistant Professor, 2012 - present
3. Non-academic experience: None
4. Certifications or professional registrations
5. Current membership in professional organizations: ACM
6. Honors and awards
ACM SIGCOMM Dissertation Award, 2013
ACM Doctoral Dissertation Award, 2012
ACM SIGCOMM Best Paper Award, 2011
ACM SIGCOMM Best Paper Award, 2008
Second AT&T Best Practical Paper Award, 2011
William A. Martin SM Thesis Award, 2008
IIT Madras Institute Award in Computer Science, 2006
7. Service activities
PC Member: 2013: SIGCOMM, MOBICOM, MOBISYS
8. Selected Publications
 - 1) Ambient Backscatter: Wireless Communication out of thin air, SIGCOMM 2013, Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, Josh Smith
 - 2) Whole-Home Gesture Recognition Using Wireless Signals, MOBICOM 2013, Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, Shwetak Patel
 - 3) They can Hear your Heartbeats: Non-Invasive Security for Wireless Medical Implants, SIGCOMM 2011, Shyamnath Gollakota, Haitham Hassaneih, Ben Ransford, Dina Katabi, Kevin Fu
 - 4) ZigZag Decoding: Combating Hidden Terminals in Wireless Networks, SIGCOMM 2008, Shyamnath Gollakota, Dina Katabi
 - 5) Secure In-Band Wireless Pairing, Usenix Security 2011, Shyamnath Gollakota, Nabeel Ahmed, Nickolai Zeldovich, and Dina Katabi
9. Briefly list the most recent professional development activities: none

Steven D. Gribble

1. Education
B.Sc., University of British Columbia, June 1995, combined Computer Science and Physics honours program
M.Sc., University of California at Berkeley, December 1997, Computer Science program.
Ph.D., University of California at Berkeley, September 2000, Computer Science program.
2. Academic experience
Professor, Department of Computer Science and Engineering, University of Washington. Full time. (9/17/2012 — present)
Associate Professor, Department of Computer Science and Engineering, University of Washington. Full time. (9/16/2005 — 9/16/2012)
Assistant Professor, Department of Computer Science and Engineering, University of Washington. Full time. (11/1/2000 – 9/15/2005)
3. Non-academic experience
Visiting Scientist, Google, Inc. Full time. (5/2008 – 4/2009)
Co-founder and acting CEO/CTO, Skytap, inc. [originally known as illumita, inc.] Full time. (6/2006-6/2007)
4. Certifications or professional registrations
5. Current membership in professional organizations: Member of USENIX and ACM.
6. Honors and awards
Torode Family Endowed Career Development Professorship in Computer Science, 2006-2008. Alfred P. Sloan Research Fellowship, 2004-2006. NSF CAREER award, 2002-2007.
Teaching Awards: Received a plaque of appreciation from my CSE451 students at the end of the Winter 2001 quarter.
7. Selected service activities
Program Committee Member: ASPLOS 2012; EuroSys 2012; OSDI 2012.
Program Co-Chair, 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2012).
Program Committee Member, 23rd ACM Symposium on Operating Systems Principles (SOSP), October 2011.
Program Committee Member, EuroSys 2011, April 2011.
Program Committee Member, 2011 IEEE Symposium on Security and Privacy (Oakland 2011), May 2011.
Program Committee Member, 19th USENIX Security Symposium, August 2010.

Program Committee Member, ACM SIGMETRICS 2010, June 2010.

8. Selected Publications

[1] Nicholas Hunt, Tom Bergan, Luis Ceze, and Steven D. Gribble. DDOS: Taming Nondeterminism in Distributed Systems. To appear in the Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2013), Houston, Texas, March 2013.

[2] David W. Richardson and Steven D. Gribble. Maverick: Providing Web Applications with Safe and Flexible Access to Local Devices. Proceedings of the 2nd USENIX Conference on Web Application Development (WebApps '11), Portland, Oregon, June 2011.

[3] Katelin Bailey, Luis Ceze, Steven D. Gribble, and Henry M. Levy. Operating Systems Implications of Fast, Cheap, Non-Volatile Memory. Proceedings of the 13th Workshop on Hot Topics in Operating Systems (HotOS XIII), Napa Valley, California, May 2011.

[4] Roxana Geambasu, John P. John, Steven D. Gribble, Tadayoshi Kohno, and Henry M. Levy. Keypad: An Auditing File System for Theft-Prone Devices. Proceedings of EuroSys 2011, Salzburg, Austria, April 2011.

[5] Tom Bergan, Nicholas Hunt, Luis Ceze, and Steven D. Gribble. Deterministic Process Groups in dOS. Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI '10), Vancouver, BC, Canada, October 2010.

[6] Jonathan M. Hsieh, Steven D. Gribble, and Henry M. Levy. The Architecture and Implementation of an Extensible Web Crawler. Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2010), San Jose, CA, April 2010.

[7] Roxana Geambasu, Steven D. Gribble, and Henry M. Levy. CloudViews: Communal Data Sharing in Public Clouds. Proceedings of the Workshop on Hot Topics in Cloud Computing (HotCloud 2009), San Diego, CA, June 2009.

[8] John P. John, Alexander Moshchuk, Steven D. Gribble, and Arvind Krishnamurthy. Studying Spamming Botnets Using Botlab. Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2009), Boston, MA, April 2009.

[9] Charles Reis and Steven D. Gribble. Isolating Web Programs in Modern Browser Architectures. Proceedings of the 4th ACM European Conference on Computer Systems (EuroSys 2009), Nuremberg, Germany, March 2009.

Dan Grossman

1. Education
Ph.D. Computer Science, Cornell University, 2003
M.S. Computer Science, Cornell University, 2001
B.S. Electrical Engineering, Rice University, 1997
B.A. Computer Science, Rice University, 1997
2. Academic experience
University of Washington, J. Ray Bowen Professor of Innovation in Engineering Education, 2013 --
University of Washington, Associate Professor, 2009 --
University of Washington, Assistant Professor, 2003 -- 2009
Harvard University, Visiting Associate Professor, 2011
3. Non-academic experience
Microsoft Research, Visiting Researcher, 2010
4. Certifications or professional registrations: None
5. Current membership in professional organizations: ACM
6. Honors and awards
University of Washington, J. Ray Bowen Professor of Innovation in Engineering Education, 2013 -- 2018
UW ACM Teacher of the Year' Award, (2008, 1st place; 2006, 2nd place; 2005 1st place)
National Science Foundation CAREER Award, 2005
Intel Graduate Student Fellowship, 2002—2003
National Science Foundation Fellowship, 1998--2001
7. Service activities
2013--present: ACM Education Board
2011--present: ACM/IEEE Computing Curricula 2013: Computer Science, Steering Committee
2010--present: ACM Education Council
2009--2012: ACM SIGPLAN Executive Committee
Over 25 conference program committees including PLDI, POPL, OOPSLA, and ASPLOS
2009 TRANSACT: 4th ACM SIGPLAN Workshop on Transactional Computing, Program Chair

8. Selected Publications

JavaUI: Effects for Controlling UI Object Access. Colin S. Gordon, Werner M. Dietl, Michael D. Ernst, Dan Grossman. European Conference on Object-Oriented Programming, Montpellier, France, July, 2013.

Rely-Guarantee References for Refinement Types over Aliased Mutable Data. Colin S. Gordon, Michael D. Ernst, Dan Grossman. ACM Conference on Programming Language Design and Implementation, Seattle, WA, June, 2013.

Identifying and Summarizing Systematic Code Changes via Rule Inference. Miryung Kim, David Notkin, Dan Grossman, Gary Wilson Jr. IEEE Transactions on Software Engineering 39(1), pages 45–62, January, 2013.

IFRit: Interference-Free Regions for Dynamic Data-Race Detection. Laura Effinger-Dean, Brandon Lucia, Luis Ceze, Dan Grossman, Hans-J. Boehm. ACM Conference on Object-Oriented Programming Systems, Languages, and Applications, Tucson, AZ, October, 2012.

RADISH: Always-On Sound and Complete Race Detection in Software and Hardware. Joseph Devietti, Benjamin P. Wood, Karin Strauss, Luis Ceze, Dan Grossman, Shaz Qadeer. 39th International Symposium on Computer Architecture, Portland, OR, June, 2012.

Type-Directed Completion of Partial Expressions. Daniel Perelman, Sumit Gulwani, Thomas Ball, Dan Grossman. ACM Conference on Programming Language Design and Implementation, Beijing, China, June, 2012.

EnerJ: Approximate Data Types for Safe and General Low-Power Computation. Adrian Sampson, Werner Dietl, Emily Fortuna, Danushen Gnanapragasam, Luis Ceze, Dan Grossman. ACM Conference on Programming Language Design and Implementation, San Jose, CA, June, 2011.

RCDC: A Relaxed-Consistency Deterministic Computer. Joseph Devietti, Jacob Nelson, Tom Bergan, Luis Ceze and Dan Grossman. 16th International Conference on Architectural Support for Programming Languages and Operating Systems, pages 67–78, Newport Beach, CA, March, 2011.

ASF: AMD64 Extension for Lock-free Data Structures and Transactional Memory. Jaewoong Chung, Luke Yen, Stephan Diestelhorst, Martin Pohlack, Michael Hohmuth, David Christie, Dan Grossman. International Symposium on Microarchitecture, Atlanta, GA, December, 2010.

9. Current membership in professional organizations: none

Bruce Hemingway

10. Education

A.B. in Music (emphasis in Electronic Music Composition and Orchestral Conducting)
Indiana University, 1973.

11. Academic experience

Senior Lecturer, Department of Computer Science & Engineering, University of Washington, 2012-present 50% FTE.

Lecturer, Department of Computer Science & Engineering, University of Washington, 2002-2012.

Manager of the [Baxter Computer Engineering Laboratory](#), 2002-present.

12. Non-academic experience

Co-founder, CEO and digital designer, dB Technologies, Inc., Seattle, WA. 1993-2002, full time.

13. Certifications or professional registrations: None

14. Current membership in professional organizations: ACM, SIAM

15. Honors and awards

16. Service activities

17. Selected Publications

Choudhury, T., Borriello, G., Consolvo, S., Haehnel, D., Harrison, B., Hemingway, B., Hightower, J., Klasnja, P. ", Koscher, K., LaMarca, A., Landay, J. A., LeGrand, L., Lester, J., Rahimi, A., Rea, A., and Wyatt, D. 2008. The Mobile Sensing Platform: An Embedded Activity Recognition System. *IEEE Pervasive Computing* 7, 2 (Apr. 2008), 32-41.

Borriello, G., Hartung, C., Hemingway, B., Koscher, K., and Mayton, B. 2008. Multi-player soccer and wireless embedded systems. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education* (Portland, OR, USA, March 12 - 15, 2008). SIGCSE '08. ACM, New York, NY, 82-86.

Bruce Hemingway, Waylon Brunette, Gaetano Borriello, "Variations on the Flock: Wireless Embedded Systems in an Undergraduate Curriculum", *Proceedings of the Australian Telecommunications, Networks and Applications Conference (ATNAC) 2006, Melbourne, Australia, December, 2006, pp. 366-370 (Invited Keynote Paper)*

Bruce Hemingway, Waylon Brunette, Tom Anderl, Gaetano Borriello. "The Flock: Mote Sensors Sing in Undergraduate Curriculum". *IEEE Computer* 37(8): 72-78 (2004)

Allen Center Art Collection Acquisition 2012 seven photographs from “Arboretum in Soft-Focus” Variable Soft-Focus Process Project series

Meany Theatre Gallery Photographic Exhibit 2011-2012 seven photographs from “Arboretum in Soft-Focus” Variable Soft-Focus Process Project series

Hall Health Center, thirteen photographs on display 2011-2013, various locations

Meany Theatre Gallery Photographic Exhibit 2010-2011 nine photographs from “Artifacts of Place” Project series

Swarms 2010 with Hugo Solis , a virtual network made of electronic sound-producing circuits positioned throughout the gallery; part of the Kirkland Arts Center exhibition, *Off the Map*, February 12 - March 10, 2010. <http://youtu.be/aJ-g6kXHfNg>.

18. Current membership in professional organizations: none

Anna R Karlin

1. Education

1981, B.Sc. in Mathematical Sciences, Stanford University.

1987, Ph.D. in Computer Science, Stanford University.

2. Academic experience

Department of Computer Science and Engineering, University of Washington, Seattle, WA.

1994-96 Visiting Associate Professor

1996-98 Associate Professor

1998 - current, Professor

2009 – current, Microsoft Professor of Computer Science and Engineering

3. Non-academic experience

1988-94, Principal Scientist (theoretical computer science research), Digital Equipment Corporation's Systems Research Center. Palo Alto, CA.

4. Certifications or professional registrations: None

5. Current membership in professional organizations: ACM, SIAM

6. Honors and awards

Selected Plenary and Distinguished Lectures:

Computation in Europe 2013: The Nature of Computation: Logic, Algorithms, Application; Women in Theory, 2010; SIAM Conference on Discrete Math, 2010; Kanellakis Distinguished Lecture, Brown University, 2008; CIS Grace Hopper Distinguished Lecturer, Univ of Pennsylvania, 2008; MIT Dertouzos Distinguished Lecturer, 2004.

ACM Fellow, 2012.

Visiting Scholar, Special Semester on Algorithmic Game Theory, Institute for Advanced Studies, Hebrew University of Jerusalem, 2011.

Outstanding Paper Award, Proceedings of ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, 1995.

7. Service activities

Chair, Faculty recruiting committee, Executive committee, Chair and member of graduate admissions committee, University of Washington CSE.

Co-organizer, MSR-UW Center Experience Theory Project, 2010, 2012.

Board Member, Computing Research Association - Women, 2012 on.

Member of External Advisory Committee: Center for Science of Information, Purdue, 2011-12.

Member of Computer Science Department Review Committee, Brown University, 2010.

Reviewer, NSF Site Review for Center for Intractability, 2010.
 Editorial Board, SIAM Journal on Computing, 1999 - 2011.
 Member, National Research Council's Computer Science and Telecommunications Board, 1999-2002.
 Member, ACM Paris Kannellakis Theory and Practice Award Subcommittee 2004-2007.
 Chair in 2006.
 Chair, ACM Paris Kanellakis Theory and Practice Award Subcommittee 2006.
 Program Chair, IEEE Symposium on Foundations of Computer Science, 1997.
 Member of numerous conference program committees.

8. Selected Publications

- Auctions for Structured Procurement, with M. Cary, A. Flaxman, and J. Hartline, In *Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.
- Improved Approximation Algorithms for Budgeted Allocations, with Y. Azar, B. Birnbaum, C. Mathieu, and C. Thach Nguyen. In *35th International Colloquium on Automata, Languages and Programming (ICALP)*, 2008.
- On the Equilibria and Efficiency of the GSP Mechanism in Keyword Auctions with Externalities, With I. Giotis, In *Workshop on Internet and Network Economics*, 2008.
- Approximating Matches Made in Heaven, with N. Chen, N. Immorlica, M. Mahdian and A. Rudra, In *36th Int. Colloquium on Automata, Languages and Programming (ICALP)*, 2009.
- Thinking Twice about Second-Price Ad Auctions, with Y. Azar, B. Birnbaum and C. Thach Nguyen. In *European Symposium on Algorithms*, 2009.
- Integrality Gaps of Linear and Semi-definite Programming Relaxations for Knapsack, with C. Mathieu and C. Thach Nguyen. In *Conference on Integer Programming and Combinatorial Optimization*, 2011.
- Prior-independent Unit-Demand Combinatorial Auctions, Nikhil Devanur, J. Hartline, A. Karlin and T. Nguyen. In *Workshop on Internet and Network Economics (WINE)*, 2011.
- Approximately Revenue-Maximizing Auctions for Deliberative Agents, with E. Celis, K. Leyton-Brown, T. Nguyen and D. Thompson, In *Conference for the Advancement of Artificial Intelligence (AAAI)*, 2012.
- Evaluating Competitive Game Balance with Restricted Play, with A. Jaffe, A. Miller, E. Andersen, Y. Liu and Z. Popovic, In *Eighth Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2012.
- Approaching Utopia: Strong truthfulness and Externality-resistant Mechanisms, with A. Fiat, E. Koutsoupias, A. Vidali. In *Innovations in Theoretical Computer Science (ITCS)*, 2013.
- Selling in Exclusive Markets: Some Observations on Prior-free Mechanism Design, with T. Nguyen, Y. Peres, In *Special Issue of Transactions on Economics and Computation*, 2013.
- On Revenue Maximization for Agents with Costly Information Acquisition, with E. Celis and D. Glekzakos, In *Int. Colloquium on Automata, Languages and Programming (ICALP)*, 2013.

9. Briefly list the most recent professional development activities: none

Ira (Irena) Kemelmacher-Shlizerman

1. Education

Ph.D., Computer Science and Applied Mathematics, Weizmann Inst. of Science, 2009

M.Sc., Computer Science and Applied Mathematics, Weizmann Inst. of Science, 2004

B.Sc., Computer Science and Mathematics, Bar-Ilan University, Israel, 2001

2. Academic experience

Assistant Professor, Computer Science and Engineering, University of Washington, 2013-now. Full time.

Research Associate (postdoc), Computer Science and Engineering, University of Washington, 2009-2013. Full time.

Research assistant, Weizmann Inst. of Science, 2004-2009. Full time.

Visiting research assistant, Columbia University, NY. Summer 2006. Full time.

3. Non-academic experience

Consultant, Google, March-Oct 2010

4. Certifications or professional registrations: n/a

5. Current membership in professional organizations: Member IEEE, ACM

6. Honors and awards

Developed the flag feature “Face Movies” of Google’s Picasa browser, 2011

Exploring Photobios” featured publication (appeared on the back cover and the trailer of SIGGRAPH), 2011

Popular press on Photobios/Face Movies in New Scientist, Discovery, KING 5 TV interview, etc.

7. Service activities (within and outside of the institution)

Keynote for [National Center for Women & Information Technology \(NCWIT\)](#), Feb 2013.

8. Selected publications

[1] I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg, S. Seitz. Exploring Photobios. *ACM Transactions on Graphics (SIGGRAPH)*, 2011

[2] I. Kemelmacher-Shlizerman, S. Seitz. Collection Flow. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012

9. Briefly list the most recent professional development activities: none

Tadayoshi Kohno

1. Education
PhD, Computer Science, UC San Diego, 2006
MS, Computer Science, UC San Diego, 2004
BS, Computer Science, University of Colorado, 1999
2. Academic experience
University of Washington, Associate Professor, 2011-present, full time
University of Washington, Assistant Professor, 2006-2011, full time
3. Non-academic experience
* Cigital, Cryptographer, cryptography and computer security consultant, 2000-2001, full time
* Counterpane Systems, Cryptographer, cryptography and computer security consultant, 1999-2000
4. Certifications or professional registrations: none
5. Current membership in professional organizations
* Senior Member, Institute of Electrical and Electronics Engineers (IEEE)
* Member, Association for Computing Machinery (ACM)
6. Honors and awards
* Defense Science Study Group (DSSG) Member, 2014
* National Science Foundation CAREER Award, 2009
* Alfred P. Sloan Research Fellowship, 2008
* Technology Review TR-35 Young Innovator Award, 2007
* Best paper awards at IEEE S&P (2012, 2008), EuroSys (2011), CPDP (2011), USENIX Security (2009), MobiSys (2008), CCS (2002), ACSAC (2000)
7. Service activities
* Conference and workshop chairmanships (LASER 2013, HUPS 2013, USENIX Security 2012, HealthSec 2010, HotSec 2009, EVT 2008)
* Conference and workshop steering committees (USENIX Security 2012-present, NDSS 2011-2014, HealthSec 2010-present)
* UW CSE Department Executive Committee (2010-2011) and Faculty Recruiting Committee (2010-2013)
* Associate Director, Technology Policy Lab, 2013-2014
8. Selected publications
* Roxana Geambasu, Tadayoshi Kohno, Amit A. Levy, and Henry M. Levy. Vanish:

Increasing data privacy with self-destructing data. In 18th USENIX Security Symposium, August 2009.

* Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive experimental analyses of automotive attack surfaces. In 20th USENIX Security Symposium, August 2011.

9. Professional development activities

Invited participant, National Effective Teaching Institute (NETI), American Society for Engineering Education, 2013

Regular meetings with the UW Center for Engineering Learning & Teaching

Arvind Krishnamurthy

1. Education

Ph.D. in Computer Science, University of California, Berkeley, May 1999.
M.S. in Computer Science, University of California, Berkeley, May 1994.
B.Tech. in Computer Science, IIT Madras, Madras, June 1991.

2. Academic experience

Associate Professor, Computer Science and Engineering, University of Washington, 2011-present.
Associate Research Professor, Computer Science and Engineering, University of Washington, 2009-2011.
Assistant Research Professor, Computer Science and Engineering, University of Washington, 2005-2009.
Assistant Professor, Computer Science Department, Yale University, 1999-2005.

3. Non-academic experience

Visiting Research Scientist, Google, 2011-2012 (part time).

4. Certifications or professional registrations: n/a

5. Current membership in professional organizations: ACM Member

6. Honors and awards

NSF Career Award
Best Paper Award, NSDI 2013
Best Paper Award, NSDI 2010
Best Paper Award, IMC 2009
Best Paper Award, NSDI 2008
Best Student Paper Award, NSDI 2007

7. Service activities

Program chair: NSDI 2015, Sigcomm 2014, NetEcon 2012, PAM 2010, IPTPS 2010.
Program-Committee Member: SOSP 2013, Sigcomm 2013, NSDI 2013, Conext 2012, CCS 2012, SigComm 2012, NSDI 2012, HotNets 2011, Conext 2011, CCS 2011, ICNP 2011, SigComm 2011, NetEcon 2011, NetDB 2011, PAM 2011, ICDCN 2011, SigComm 2010, NetEcon 2010, Sigmetrics 2010, ICDCS 2010, HotNets 2009, SOSP 2009, NSDI 2009, IPTPS 2009, HotWeb 2008, IMC 2008, IMC 2007.

Tutorials chair for Sigmetrics '09.
GENI distributed services working group, 2006-2008.

Coordinator of Undergraduate Direct Admits Seminar, 2012
Undergraduate Scholarships Committee, 2012
Undergraduate Admissions Committee, 2012
Coordinator of Undergraduate Direct Admits Seminar, 2011

8. Briefly list the most important publications and presentations from the past five years

V. Liu, D. Halperin, A. Krishnamurthy, and T. Anderson, "F10: Fault Tolerant Engineered Networks," in Proceedings of Networked Systems Design and Implementation, 2013. (Award Paper.)
S. Han, V. Liu, Q. Pu, S. Peter, T. Anderson, A. Krishnamurthy, and D. Wetherall, "Expressive Privacy Control with Pseudonyms," in Proceedings of SIGCOMM, 2013.
U. Javed, Í. Cunha, D. Choffnes, E. Katz-Bassett, T. Anderson, and A. Krishnamurthy, "PoiRoot: Investigating the Root Cause of Interdomain Path Changes," in Proceedings of SIGCOMM, 2013.
E. Katz-Bassett, C. Scott, D. Choffnes, I. Cunha, V. Valancius, N. Feamster, H. Madhyastha, T. Anderson, and A. Krishnamurthy, "LIFEGUARD: Practical Repair of Persistent Route Failures," in Proceedings of SIGCOMM, 2012.
J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service," in Proceedings of SIGCOMM, 2012.
L. Glendenning, I. Beschastnikh, A. Krishnamurthy, and T. Anderson, "Scalable Consistency in Scatter," in Proceedings of Symposium on Operating System Principles, 2011.
E. Katz-Bassett, H. Madhyastha, V. Adhikari, C. Scott, J. Sherry, P. van Wessep, T. Anderson, and A. Krishnamurthy, "Reverse Traceroute", in Proceedings of Networked Systems Design and Implementation, 2010. (Award Paper.)
R. Krishnan, H. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao, "Moving Beyond End-to-End Path Information to Optimize CDN Performance," in Proceedings of Internet Measurement Conference, 2009. (Award paper.)
J. John, E. Katz-Bassett, A. Krishnamurthy, T. Anderson, and A. Venkataramani, "Consensus Routing: the Internet as a Distributed System," in Proceedings of Networked Systems Design and Implementation, 2008. (Award Paper.)

9. Briefly list the most recent professional development activities: none

Edward D. Lazowska

1. Education

Ph.D. in Computer Science, Univ. of Toronto, 1977

M.Sc. in Computer Science, Univ. of Toronto, 1974

A.B. in Computer Science (independent concentration), Brown Univ., 1972

2. Academic experience

Univ. of Washington, Dept. of Computer Science & Engineering

Bill & Melinda Gates Chair, 2000-present

Department Chair, 1993-2001

Professor, 1986-present; Associate Professor, 1982-86; Assistant Professor, 1977-82

Univ. of Washington, eScience Institute

Founding Director, 2008-present

Univ. of Washington, School of Oceanography

Adjunct Professor, 2012-present

3. Non-academic experience

Digital Equipment Corp., Systems Research Center

Visiting Scientist, 1984-85

Quantitative System Performance

Co-Founder and President, 1982-2007

Served on the Microsoft Research Technical Advisory Board since its inception in 1991; serve as a board member or technical advisor to a number of high-tech companies and venture firms

4. Certifications or professional registrations: None

5. Current membership in professional organizations: See (7)

6. Honors and awards

Member of the National Academy of Engineering

Fellow of the American Academy of Arts & Sciences

Member of the Washington State Academy of Sciences

Fellow of the Association for Computing Machinery

Fellow of the Institute of Electrical and Electronic Engineers

Fellow of the American Association for the Advancement of Science

Vollum Award for Distinguished Accomplishment in Science and Technology, Reed College, 2012

Inaugural ACM SIGMETRICS Test-of-Time Award, 2010

ACM Distinguished Service Award, 2009

University of Washington Computer Science & Engineering Undergraduate Teaching Award, 2007

Association for Computing Machinery Presidential Award, 2005

- Computing Research Association Distinguished Service Award for outstanding service to the computing research community, 2005
- University of Washington Outstanding Public Service Award, 1998
- University of Washington Annual Faculty Lecturer, 1996
- Award papers: ACM SIGCOMM '93 Symposium, 1993 Machnix Workshop, 13th ACM Symposium on Operating Systems Principles (1991), 12th ACM Symposium on Operating Systems Principles (1989), 1989 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, 1985 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, 7th ACM Symposium on Operating Systems Principles (1979)
7. Service activities (within and outside of the institution)
- External (major, past 10 years):
- Institute for Systems Biology, President's Council, 2012-
 - NASA Advisory Council, Information Technology Infrastructure Committee, 2012-
 - Co-chair (with D.E. Shaw), Working Group of the President's Council of Advisors on Science and Technology (PCAST) to review the Networking and Information Technology Research and Development (NITRD) Program, 2010
 - Chair, Computing Community Consortium, 2007-
 - U.S. Department of Energy, Energy Efficiency and Renewable Energy Advisory Committee (ERAC), 2010-12
 - U.S. Department of Energy Pacific Northwest National Laboratory Fundamental & Computational Sciences Directorate Advisory Committee, 2009-
 - National Research Council Committee on Management of University Intellectual Property, 2008-10
 - Co-chair (with M. Benioff), President's Information Technology Advisory Committee, 2003-05
 - Chair, Information Science and Technology (ISAT) Study Group of the Defense Advanced Research Projects Agency, 2005-06; Vice Chair (Chair-designate), 2003-04; Member, 1998-2001
8. Selected publications and presentations
- Distinguished Lectures in the past three years: University of Michigan, Harvard University, Boston University, University of Waterloo, Modern Language Association, MIT 150th Anniversary Celebration, Consortium for Computing Sciences in Colleges, Computing Research Association; testimony twice to the House Science Committee.
9. Briefly list the most recent professional development activities: none

James R. Lee

1. Education –
PhD Computer Science, University of California, Berkeley, 2005
B.S. Computer Science & Mathematics, Purdue University, 2001
2. Academic experience –
Visiting Associate Professor, Computer Science, U. C. Berkeley, 2012-2013
Associate Professor, Computer Science & Engineering, U. Washington, 2011-present.
Assistant Professor, Computer Science & Engineering, U. Washington, 2006-2011
Postdoctoral Associate, Institute for Advanced Study, Princeton, 2005-2006.
3. Non-academic experience -
Consultant, Microsoft Research, 2006-present
Intern, IBM Research, 2004
Intern, Microsoft Research, 2003 & 2004
4. Certifications or professional registrations
5. Current membership: ACM, IEEE Computer Society
6. Honors and awards:
Sloan Research Fellowship, 2009
NSF CAREER Award, 2007
Institute for Advanced Study 2-year Fellowship, 2004
NSF Graduate Research Fellowship, 2001
7. Selected service activities:
National and international:
Organizer, “Algorithmic Spectral Graph Theory” at the Simons Institute for Theoretical Computer Science, 2014.
Organizer, “Workshop on L_1 embeddings and flow/cut gaps,” in conjunction with the ASM Symposium on Computational Geometry, 2012.
Organizer, “Metric geometry, algorithms, and groups,” Institute Henri Poincare (Paris), 2011.
Organizer, “The Experience Theory Project,” (summer program for undergrad and grad students), 2010 and 2012.
Program committee member for SODA, FOCS, STOC.
University of Washington Service:
Executive Committee, 2009 and 2013

8. Selected Publications:

Siu-On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large LP relaxations. To appear, *Proceedings of the 54th Annual Symposium on the Foundations of Computer Science*, 2013.

James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multi-way spectral partitioning and higher-order Cheeger inequalities. In *Proceedings of the 44th Annual Symposium on the Theory of Computing*, 2012.

Jian Ding, James R. Lee, and Yuval Peres. Cover times, blanket times, and majorizing measures. *Annals of Mathematics*, 175(3): 1409—1471, 2012.

P. Biswal, James R. Lee, and Satish Rao. Eigenvalue bounds, spectral partitioning, and metrical deformation via flows. *Journal of the Association for Computing Machinery*, 57(3): 13(1—23), 2010.

Sanjeev Arora, James R. Lee, and Assaf Naor. Euclidean distortion and the Sparsest Cut. *Journal of the American Mathematical Society*, 21(1): 1—21, 2008.

Henry M. Levy

1. Education

MS, Computer Science, University of Washington, 1981
BS, Math/Computer Science, Carnegie Mellon University, 1974

2. Academic experience

Current Position. Professor, Chairman, and Wissner-Slivka Chair in Computer Science & Engineering, Department of Computer Science and Engineering, University of Washington (Assistant Prof., 1983-1988; Associate Prof. 1988-1994; Prof. 1994; Microsoft Professor, 1999; Wissner-Slivka Chair, 2004; Associate Chairman, 1998 – 2005, Chairman 2006). Research in operating systems, computer architecture, multithreaded computer systems, object-oriented systems and languages, performance evaluation, computer security, the Internet and World-Wide Web.

3. Non-academic experience

Consulting Engineer, Digital Equipment Corporation, 1974 – 1983

4. Certifications or professional registrations: None

5. Current membership in professional organizations: Fellow of the IEEE, Fellow of the ACM, Member of the National Academy of Engineering

6. Honors and awards

Member of the National Academy of Engineering, 2011; Wissner-Slivka Endowed Chair in Computer Science and Engineering, 2004; Microsoft Endowed Professorship in Computer Science and Engineering, 1999; Fellow of the Institute for Electrical and Electronics Engineers (IEEE), 2002; Fellow of the Association for Computing Machinery (ACM), 1995; Fulbright Research Scholar Award (France, 1992);

Award Papers and Paper Honors (19 total):

ACM European Systems Conference (Eurosys 2011)

International Symposium on Computer Architecture (Test of Time Award, 2010, 2011)

ACM Symposium on Operating Systems Principles (Test of Time Award, 2009)

USENIX Security Conference (2009)

Conf. on Mobile Systems, Applications, and Services (2008)

IEEE Symposium on Security and Privacy (2006)

13th International Conference on the World-Wide Web (2004)

ACM Symposium on Operating Systems Principles (1985, 1987, 1989, 1991, 1999, 2003)

USENIX/ACM Symposium on Operating Systems Design and Implementation (2002, 2004)

USENIX Conf. on Internet Technologies and Systems (2001)

International Conference on Measurement and Modeling of Computer Systems (1989)

Third European Conference on Object-Oriented Programming (1989)

7. Service activities (within and outside of the institution)
 - Program Co-Chair, USENIX Symp. On Operating Systems Design and Implementation, 2014.
 - Program Committee, USENIX Symp. On Operating Systems Design and Implementation, October 2010.
 - Program Committee, 21st ACM Symp. On Operating Systems Principles, October 2007.
 - Program Committee, 11th Int. Conf. on Arch. Support for Prog. Languages and Operating Sys., October 2004.
 - Program Committee, 4th USENIX Symp. on Internet Technologies and Systems, March 2003.
 - Program Committee, 10th Int. Conf. on Arch. Support for Prog. Languages and Operating Sys., October 2002.
 - Program Committee, 6th International Symposium on High Performance Computer Architecture, January 2000.
8. Selected Publications
 - Computer Security and the Modern Home, with Tamara Denning and Tadayoshi Kohno. Communications of the ACM 56(1), January 2013.
 - Keypad: An Auditing File System for Theft-Prone Devices, with R. Geambasu, J.P. John, T. Kohno, and S. D. Gribble. Proc. of the European Conf. on Computer Systems, April 2011 (Award paper).
 - Vanish: Increasing Data Privacy with Self-Destructing Data, with R. Geambasu, T. Kohno, and A. Levy. Proc. of the 18th USENIX Security Symposium, August 2009 (Award Paper).
 - Organizing and Sharing Distributed Personal Web-Service Data, with R. Geambasu, C. Cheung, A. Moshchuk, and S. Gribble. Proc. of the 17th International World Wide Web Conference (WWW 2008), April 2008.
 - Flashproxy: Transparently Enabling Rich Web Content via Remote Execution, with A. Moshchuk and S. Gribble. Proc. of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys 2008), June 2008 (honorable mention paper).
9. Briefly list the most recent professional development activities: None

Hélène Martin

1. Education
B.S. Computer Science, B.A. Linguistics, June 2008, University of Washington
2. Academic experience
University of Washington, Lecturer, 2011-present, full time
3. Non-academic experience
Garfield High School, Computer Science Teacher, 2009-2011, full time
Grameen Technology Center, Technology Innovation Consultant, 2008-2009, full time
4. Certifications or professional registrations
5. Current membership in professional organizations
Association for Computing Machinery
Computer Science Teachers Association
6. Honors and awards
Yale Educator Award, 2012
UW Computer Science and Engineering Inspirational Teacher Award, 2011
MIT Influential Teacher, 2010
Bob Bandes Memorial Award for Excellence in Teaching, 2007
7. Service activities (within and outside of the institution)
Advanced Placement CS Exam Reader, 2011, 2013
Business/Marketing/IT Advisory Group for Seattle Public Schools, 2012 - present
UW in the High School CSE 142 liaison, 2011 – present
Association for Computing Machinery for Women group advisor, 2012 - present
Puget Sound CS Teachers Association planning committee, 2009 – present
8. Briefly list the most important publications and presentations from the past five years – title, co-authors if any, where published and/or presented, date of publication or presentation
Broadening Participation: The Why and the How. IEEE Computer. February 2013.
9. Briefly list the most recent professional development activities: none.

Mausam

1. Education
PhD, Computer Science, University of Washington, 2007
MS, Computer Science, University of Washington, 2004
BTech, Computer Science, Indian Institute of Technology Delhi, 2001
2. Academic experience
Research Assistant Professor, University of Washington (2007-present)
3. Non-academic experience
4. Certifications or professional registrations
5. Current membership in professional organizations
Member, Association for the Advancement of Artificial Intelligence (AAAI)
Member, Association for Computational Linguistics (ACL)
6. Honors and awards
Honorable Mention for the 2008 ICAPS Best Dissertation Award for the best dissertation in planning and scheduling for the year 2007.
Runners-up in International Probabilistic Planning Competition 2011, MDP Track, to our planner Glutton. (Team: Andrey Kolobov, Peng Dai, Mausam, Daniel S. Weld.)
Recipient of the silver medal for being first rank in the computer science department at IIT in the year 2000-2001.
7. Service activities (within and outside of the institution)
Local Chair: AAAI Conference on Artificial Intelligence (AAAI'13), Symposium on Combinatorial Search (SoCS'13), Symposium on Abstraction, Reformulation, and Approximation (SARA'13).
Area Chair: Empirical Methods in Natural Language Processing (EMNLP'13).
Senior Program Committee Member: International Conference on Automated Planning and Scheduling (ICAPS'13), AAAI Conference on Artificial Intelligence (AAAI'10, AAAI'11), International Joint Conference on Artificial Intelligence (IJCAI'11, IJCAI'13).
Program Committee Member: International Joint Conference on Artificial Intelligence (IJCAI'09), AAAI Conference on Artificial Intelligence (AAAI'05, AAAI'08), International Conference on Automated Planning and Scheduling (ICAPS'07, ICAPS'08, ICAPS'09, ICAPS'10, ICAPS'11), European Chapter of the Association for Computational Linguistics (EACL'12), Empirical Methods in Natural Language Processing (EMNLP'10, EMNLP'12), Annual Meeting of the Association for Computational Linguistics (ACL'11), Conference on Knowledge Discovery and Data Mining (KDD'13), Conference on Human Computation & Crowdsourcing (HCOMP'13), Workshop on Information Extraction &

Entity Analytics on Social Media Data at COLING'12, Workshop on International Planning Competition at ICAPS'12, AAAI Student Abstracts (AAAI'11, AAAI'13), Workshop on Foundations and Methodologies for Learning by Reading at IJCAI'11, Workshop on Decision Making in Partially Observable, Uncertain Worlds at IJCAI'11, **Panel Reviewer:** National Science Foundation, Division of Information & Intelligent Systems (2008, 2010,2013).

Award Selection Committee: ICAPS Most Influential Paper (10 years older) and Best Dissertation Awards (2011).

Session Chair: AAAI Conference on Artificial Intelligence (AAAI'10, AAAI'11), Conference on Empirical Methods in Natural Language Processing (EMNLP'12), Workshop on Foundations and Methodology for Learning by Reading at NAACL'10.

Mentor: ICAPS'10 Doctoral Consortium.

8. Selected Publications

Mausam, Andrey Kolobov. "Planning with Markov Decision Processes: An AI Perspective". Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers. June 2012.

Andrey Kolobov, Mausam. "MDPs for Probabilistic Planning". A tutorial at AAAI Conference on Artificial Intelligence (AAAI). Toronto, ON, Canada. July 2012.

Andrey Kolobov, Mausam, Daniel S. Weld. "Discovering Hidden Structure in Factored MDPs". Artificial Intelligence Journal (AIJ). Volume 189 (2012) 19-47.

Alan Ritter, Sam Clark, Mausam, Oren Etzioni. "Named Entity Recognition in Tweets: An Experimental Study". Conference on Empirical Methods in Natural Language Processing (EMNLP). Edinburgh, UK. July 2011.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, Mausam. "Open Information Extraction: the Second Generation". International Joint Conference on Artificial Intelligence (IJCAI). *Invited paper*. Barcelona, Spain. July 2011.

Peng Dai, Mausam, Daniel S. Weld. "Decision-Theoretic Control of Crowd-Sourced Workflows". AAAI Conference on Artificial Intelligence (AAAI). Atlanta, GA, USA. July 2010.

Alan Ritter, Mausam, Oren Etzioni. "A Latent Dirichlet Allocation method for Selectional Preferences". Annual Meeting of the Association for Computational Linguistics (ACL). Uppsala, Sweden. July 2010.

Mausam, Stephen Soderland, Oren Etzioni, Daniel S. Weld, Kobi Reiter, Michael Skinner, Marcus Sammer, Jeff Bilmes. "Panlingual Lexical Translation via Probabilistic Inference". Artificial Intelligence Journal (AIJ). Volume 174 Issues 9-10 (2010) 619-637.

Barbara Mones

1. Education
Post Grad Animation Cert, Sheridan College, Ontario Canada.
Rhode Island School of Design, MFA
University of Michigan, BFA
2. Academic Experience
Creative Director, Animation Research Labs and Senior Lecturer, Computer Science & Engineering, University of Washington, Seattle (1999 -13)
Visiting Faculty, University of Tasmania, Australia, January.2009
Erskine Research Fellow, HITlab, (Human Interface Technologies Lab), University of Canterbury, Christ Church, New Zealand, 2006-2007,
Associate Professor and Founding Director, Visual Information Technologies MA/MFA Graduate Program
George Mason University, Fairfax, Virginia 1992-1999
3. Non-academic Experience
2006-2007 Erskine Research Fellow, HITLab , University of Canterbury, Fulltime, Christchurch NZ.
1997-1999 Animation/Technical Writer/Designer, Fulltime, Industrial Light and Magic, San Rafael. CA,
Animation/Technical Writer/Trainer/Production Artist 1996-1997, Pacific Data Images, Palo Alto, CA
4. Certifications or professional registrations: SIGGRAPH Member
5. Current membership in professional organizations: SIGGRAPH, ASIFA
6. Honors & Awards
NASA Group Achievement Award, GLOBE Project, Film Screening awards
Nominations Committee (for Executive Committee) SIGGRAPH, UWTV Backstory Interview. "Making Of "Footage and Film Screenings online:
<http://uwtv.org/series/backstory/> Multiple film screenings on campus., proposed and taught six new courses since I arrived in 1999. **Intel Science and Technology Center for Visual Computing (ISTC-VC) PI**, Two Themes; Content Development and Simulation of Virtual Characters.
7. Service Activities
Screened (competitively) thirteen films created and produced here in CSE all over the world. Have won several awards: <http://www.cs.washington.edu/research/ap/films.html>

Liaison to DXarts Program (UW), Liaison to Cinema Studies Program. Coordinated student visits to and from major animation production houses in California, Oregon and Canada. Many students from this program are now professionals in the animation industry thanks to our program in CSE. This year one of the students who graduated from CSE animated on “Paper Man”, a short animated film from Disney that won the academy award in that area. 2012 Disney Research, Facial Expressions and the Uncanny Valley (Zurich, Switzerland) and Fast Prototyping for Animation, (Saarland University, Germany), September, 2012.

8. Selected Publications

“The Cinematic Sandbox: Fast Prototyping for Digital Animation” and “Determining Facial Expression Triggers for Stylized Characters” 2010-present
Barbara Mones, Gary Faigin, Ira Kemelmacher, Charles Dickey, “Data Driven Fundamentals for Facial Expressions in Stylized Characters” ” Submitted to technical papers program, SIGGRAPH, 2013.

9. Professional Development Activities

Collaboration with Cinema Studies on an undergraduate major titled “New Media.”
Development of an online (MOOC) course for CSE. Supervising and designing and implementing new software tools for fast prototyping animation and applying it into our production curriculum.

Mark Oskin

1. Education
 - 1996 B.S. in Computer Science, University of California at Davis
 - 2000 M.S. in Computer Science, University of California at Davis
 - 2001 Ph.D. in Computer Science, University of California at Davis
 - Thesis: *Active Pages: A Computational Model for Intelligent Memory*
2. Academic Experience
 - Assistant Professor, Computer Science and Engr, Univ. of Washington, 2001-2007
 - Associate Professor, Computer Science and Engr, Univ. of Washington, 2007-Present
 - Visiting Faculty, Universitat Politècnica de Catalunya 2011
3. Non-academic Experience
 - Microsoft Visiting Faculty & Consultant, July 2006-2008
 - Co-Founder, Chief Technology Officer, Corensic, 2008-2012
 - Co-Founder, VP of Engineering, Konyac, 2013-
4. Certifications or professional registrations (None)
5. Current membership in professional organizations: ACM Member
6. Honors and Awards
 - 2001 NSF CAREER Award
 - 2005 Sloan Research Fellowship
7. Service Activities
 - July 2008, Dec 2010, Dec 2011, Jan 2012, April 2012, April 2013 NSF Panel Review Member
 - 2008 Tutorial organizer: RAMP
 - 2008 ASPLOS Local arrangements co-chair
 - 2009 Program committee member for the International Symposium on Computer Architecture (ISCA)
 - 2009-2010 Co-Organizer (along with Josep Torrellas) of CCC Workshop on Advancing Computer Architecture Research.
 - 2011, 2012 Program committee member for SIGMETRICS
 - 2012 Program committee member for SIGMETRICS
 - 2012 Program committee member for IEEE Top Picks
 - 2013 Program committee member for ASPLOS
 - 2013 Program committee member for HPCA
 - 2013 Program committee member for HotPar
 - 2013 Workshop and Tutorial Chair for MICRO-46
 - 2011 - Present, Associated Editor for Computer Architecture Letters
 - 2007-08 Co-director of the Experimental Computer Engineering Lab (ExCEL)
 - 2008-09 Co-director of the Experimental Computer Engineering Lab (ExCEL)
 - 2011-12 Professional Masters Admissions Committee
 - 2012-13 Co-director of the Experimental Computer Engineering Lab (ExCEL)
8. Selected Publications
 - Martha Mercaldi-Kim, John D. Davis, Mark Oskin, Todd Austin, "Polymorphic On-chip

Networks”, *International Symposium on Computer Architecture (ISCA)* 2008, (10 pages).

Acceptance Rate: 14%

Lucas Kreger-Stickles and Mark Oskin, “Microcoded Architectures for Ion-trap Quantum Computers”, *International Symposium on Computer Architecture (ISCA)* 2008, (10 pages).

Acceptance Rate: 14%

Joseph Devietti, Brandon Lucia, Luis Ceze and Mark Oskin, “Explicitly Parallel Programming with Shared-Memory is Insane: At Least Make it Deterministic!”, *Workshop on Software and Hardware Challenges for Many Core Platforms, held in conjunction with ISCA* 2008, (10 pages).

Joseph Devietti, Brandon Lucia, Luis Ceze and Mark Oskin, “DMP: Deterministic Shared Memory Multiprocessing”, *Architectural Support for Programming Languages and Operating Systems (ASPLOS)* 2009, (12 pages). *Acceptance Rate: 26%*. Also chosen for IEEE Top Picks 2009.

Jacob Nelson, Brandon Myers, A. H. Hunter, Preseton Briggs, Luis Ceze, Carl Ebeling, Dan Grossman, Simon Kahan, and Mark Oskin, “Crunching Large Graphs with Commodity Processors”, *USENIX HotPar* 2011.

9. Professional development activities (None)

Shwetak Patel

1. Education
Ph.D., Computer Science, Georgia Institute of Technology, 2008
B.S., Computer Science, Georgia Institute of Technology, 2003
2. Academic experience
University of Washington, Assistant Professor, 2008-current, full time
3. Non-academic experience
Zensi, Inc., Co-founder, 2007-2010, part time
Belkin Inc., Consultant, 2010-current, part time
Microsoft Research, Consultant, 2010-current, part time
SNUPI Inc., Co-founder, 2012-current, part time
4. Certifications or professional registrations
None
5. Current membership in professional organizations
ACM
IEEE
6. Honors and awards
NSF Career Award (2013)
Sloan Fellowship (2012)
MacArthur Fellowship (2011)
Microsoft Research Faculty Fellowship (2011)
College of Engineering Community Faculty Innovator Award (2011)
Seattle Magazine Most Influential (2011)
Georgia Tech GVU Center 20 Year Impact Award
Newsmakers of the Year by TechFlash and Seattle Business Journal (2010)
TR-35 Award (2009)
7. Service activities (within and outside of the institution)
UW EE Faculty Search Committee (2012, 2013)
UW CSE ACM Committee (2011)
UW Green IT Certificate Program Advisory Board (2011)
HKN EE/CSE Honor Society Advisor (2011, 2012)
UW EE Undergraduate Research Committee (2009, 2010)
Advisor for UW Smart Grid Initiative (2010)

House Science Committee Briefing (2013)

Briefed Secretary of Energy Steven Chu (2011)
Briefed Department of Energy Agency Directors on Residential Energy Monitoring (2010-2011)
Briefed and advised Chairman Jon Wellen of the Federal Energy Regulatory Commission (August 2009)
Briefed and advised Seattle City Light and Seattle Public Utilities on Clean Technology

8. Most important publications and presentations from the past five years
Larson, E.C., Goel, M., Boriello, G., Heltshe, S., Rosenfeld, M., Patel, S.N. *SpiroSmart: Using a Microphone to Measure Lung Function on a Mobile Phone*. In the Proceedings of the ACM International Conference on Ubiquitous Computing (UbiComp 2012). pp. 280-289. 2012. (Acceptance Rate: 19%) (**Best Paper Award Honorable Mention**).

Cohn, G., Lee, T., Gupta, S., Morris, D., Smith, J.R., Reynolds, M.S., Tan, D.S., Patel, S.N. *An Ultra-Low-Power Human Body Motion Sensor Using Static Electric Field Sensing*. In the Proceedings of the ACM International Conference on Ubiquitous Computing (UbiComp 2012). pp. 99-102. 2012. (Acceptance Rate: 19%) (**Best Paper Award**).

Cohn, G., Morris, D., Patel, S.N., Tan, D.S. *Your Noise is My Command: Sensing Gestures Using the Body as an Antenna*. CHI 2011. ACM Conference on Human Factors in Computing Systems (CHI 2011). pp. 791-800. 2011. (Acceptance Rate: 25%) (**Best Paper Award**)

Gupta, S., Reynolds, M.S., Patel, S.N. *ElectriSense: Single-Point Sensing Using EMI for Electrical Event Detection and Classification in the Home*. In the Proceedings of the ACM International Conference on Ubiquitous Computing (UbiComp 2010). pp. 139-148. 2010. (Acceptance Rate: 19%) (**Best Paper Award**)

Cohn, G., Stuntebeck, E., Pandey, J., Otis, B., Abowd, G.D., Patel, S.N. *SNUIP: Sensor Nodes Utilizing Powerline Infrastructure*. In the Proceedings of the International Conference on Ubiquitous Computing (UbiComp 2010). pp. 159-168. 2010. (Acceptance Rate: 19%) (**Best Paper Award Honorable Mention**)

Patel, S.N., Gupta, S., Reynolds, M. *The Design and Evaluation of an End-User-Deployable, Whole House, Contactless Power Consumption Sensor*. ACM Conference on Human Factors in Computing Systems (CHI 2010). pp. 2471-2480. 2010. (Acceptance Rate: 22%) (**Best Paper Award Honorable Mention**)

Froehlich, J. Larson, E., Campbell, T., Haggerty, C., Fogarty, J., and Patel, S.N. *HydroSense: Infrastructure-Mediated Single-Point Sensing of Whole-Home Water Activity*. In the Proceedings of UbiComp 2009. pp. 235-244. (Acceptance Rate: 12%) (**Nominated for Best Paper Award**)

Hal Perkins

1. Education
M.S., Computer Science, Cornell University, 1982
B.S., Mathematics, Arizona State University, 1975.
2. Academic Experience
University of Washington, Sr. Lecturer, Computer Science & Engineering, 2001-present, full time
University of Washington, Lecturer, CSE, 1998-2001, full time
Cornell University, Lecturer, Computer Science, 1983-1987, 1993-1998, full time
Cornell University, Research Staff, Computer Science, 1987-1993, full time
3. Non-academic Experience
AiResearch Mfg. Co., Phoenix, AZ, Engineering Associate, staff programmer for engineering fluid dynamics simulation project, 1975-1976, full time
AiResearch Mfg. Co., Phoenix, AZ, Systems Programmer, support staff for scientific computer center, 1974-1975, part time
4. Certifications or professional registrations: n/a
5. Current membership in professional organizations: Member, ACM; Member, IEEE
6. Honors & Awards
2011-12 UW ACM Teaching Award
7. Service Activities
College of Engineering Council on Educational Policy, 2010-present
UW CSE ACM chapter faculty advisor, 2010-present
UW CSE PMP admissions committee, 2004-2012
UW CSE Non-Majors curriculum committee, 2011-12
8. Selected Publications: n/a
9. Briefly list the most recent professional development activities: n/a

Anup Rao

1. Education –
PhD Computer Science, University of Texas, 2007
B.Sc. Computer Science, Georgia Institute of Technology, 2002
B.Sc. Mathematics, Georgia Institute of Technology, 2002
2. Academic experience –
Assistant Professor, Computer Science & Engineering, U. Washington, 2010–Present.
Research Scientist, Princeton University, 2009–10
Member, Institute for Advanced Study, Princeton, 2007–09
3. Non-academic experience -
Consultant, Microsoft Research 2001-02
4. Certifications or professional registrations
5. Current membership: ACM
6. Honors and awards:
Sloan Research Fellowship 2011
NSF Career Award 2011
7. Selected service activities:
Program Committee: ACM Symposium on Theory of Computation, 2009
Program Committee: IEEE Conference on Computational Complexity, 2011
8. Selected Publications:

Anup Rao.

Parallel Repetition in Projection Games and a Concentration Bound
SICOMP Special Issue for STOC 2008.

Anup Rao,

Extractors for a Constant Number of Polynomially Small Entropy Independent
Sources *SICOMP Special Issue for STOC 2006*.

Jesse Kamp, Anup Rao, Salil Vadhan, David Zuckerman.

Deterministic Extractors for Small Space Sources
Journal of Computer and System Sciences, 2010.

Boaz Barak, Mark Braverman, Xi Chen, Anup Rao.

How to Compress Interactive Communication
SICOMP Special Issue for STOC 2010.

Walter L. Ruzzo

1. Education

BS, Mathematics, California Institute of Technology, 1968

Ph.D., Computer Science, University of California, Berkeley, 1978

2. Academic experience – (all full time)

1977-1982 Asst Prof Computer Science, University of Washington

1984-1985 Visiting Scholar, EECS, University of California, San Diego

1982-1990 Assoc Prof Computer Science, University of Washington

1990-1991 Visiting Prof Computer Science, University of Toronto

2002-2003 Visiting Prof, Institute for Systems Biology, Seattle, WA

1990- Prof Computer Science and Engineering, University of Washington

2001- Adjunct Professor of Genome Sciences, University of Washington

2008- Joint Member, Fred Hutchinson Cancer Research Center

3. Non-academic experience

1966-1973 Professional Programmer and Systems Analyst

4. Certifications or professional registrations: None

5. Current membership in professional organizations: None

6. Honors and awards

7. Service activities (within and outside of the institution)

8. Selected Publications

Books

Gorodkin, Ruzzo (eds.) *RNASequence, Structure and Function: Computational and Bioinformatic Methods*. Methods in Mol Bio. Humana Press, To Appear, 2013. (Springer Protocols).

Refereed Journals

Bar, Wyman, Fritz, Qi, Garg, Parkin, Kroh, Bendoraite, Mitchell, Nelson, **Ruzzo**, Ware, Radich, Gentleman, Ruohola-Baker, Tewari. "MicroRNA Discovery and Profiling in Human Embryonic Stem Cells by Deep Sequencing of Small RNA Libraries." *Stem Cells*. 26(10) 10/2008, 2496-05. [PMID: 18583537](#)

Anandam, Torarinsson, **Ruzzo**. "Multiper: shuffling multiple sequence alignments while approximately preserving dinucleotide frequencies." *Bioinformatics* 25, 668-669, 2009. [PMID: 19136551](#).

Tseng, Weinberg, Gore, Breaker, **Ruzzo**. "Finding non-coding RNAs through genome-scale clustering." *J Bioinform Comput Biol*. 2009 Apr;7(2):373-88. [PMID: 19340921](#)
[PMC3417115](#)

Gorodkin, Hofacker, Torarinsson, Yao, Havgaard, **Ruzzo** “De novo prediction of structured RNAs from genomic sequences.” (Feature Review) *Trends in Biotechnology* 28, 2010.

[PMID: 19942311](#)

Cao, Yao, Sarkar, Lawrence, Sanchez, Parker, MacQuarrie, Davison, Morgan, **Ruzzo**, Gentleman, Tapscott. “Genome-wide MyoD binding in skeletal muscle cells: a potential for broad cellular reprogramming.” *Developmental Cell*. 2010 vol. 18 (4) pp. 662-74. [PMID: 20412780](#) [PMC2910615](#)

Knouf, Garg, Arroyo, Correa, Sarkar, Parkin, Wurz, O'Briant, Godwin, Urban, **Ruzzo**, Gentleman, Drescher, Swisher, Tewari. "An integrative genomic approach identifies p73 and p63 as activators of miR-200 microRNA family transcription." *Nucleic Acids Res*. 2012 Jan 1;40(2):499-510. [PMID 21917857](#)

Geng, Yao, Snider, Fong, Cech, Young, van der Maarel, **Ruzzo**, Gentleman, Tawil, Tapscott “DUX4 Activates Germline Genes, Retroelements, and Immune Mediators: Implications for Facioscapulohumeral Dystrophy.” *Developmental Cell*. 2012 Jan 17; 22(1):38-51. [PMID: 22209328](#) [PMC3264808](#)

Jones, **Ruzzo**, Peng, Katze. “A new approach to bias correction in RNA-Seq.”

Bioinformatics 2012 Apr 1;28(7):921-8. [PMID: 22285831](#)

Fong, Yao, Zhong, Cao, **Ruzzo**, Gentleman, Tapscott. “Genetic and epigenetic determinants of neurogenesis and myogenesis.” *Developmental Cell*. 2012 Apr 17;22(4):721-35. [PMID: 22445365](#)

Seemann, Sunkin, Hawrylycz, **Ruzzo** and Gorodkin. “Transcripts with in silico predicted RNA structure are enriched everywhere in the mouse brain.” *BMC Genomics* 2012 May 31; 13:214. [PMID: 22651826](#) [PMC3464589](#)

Jones, **Ruzzo**, Peng, Katze. “Compression of next-generation sequencing reads aided by highly efficient de novo assembly.” *Nucleic Acids Res*. 2012 Dec 1;40(22):e171 [PMID: 22904078](#) [PMC3526293](#)

Bolouri, **Ruzzo** “Integration of 198 ChIP-seq datasets reveals human cis-regulatory regions.” *J Comp Biol*, 2012 Sep;19 (9):989-997. [PMID: 22897152](#)

Tseng, Hullar, Li, Lampe, Sandstrom, Johnson, Strate, **Ruzzo**, Stamatoyannopoulos.” A microbial profiling method for the human microbiota using high-throughput sequencing.” *Metagenomics*, 2:8, 2013.

Yao, Fong, Cao, **Ruzzo**, Gentleman, Tapscott. “Comparison of endogenous and overexpressed MyoD shows enhanced binding of physiologically bound sites.” *Skeletal Muscle*, 3(1):8, Apr. 2013. [PMID:23566431](#).

Refereed Conferences

Olson, Kim, Clauson, Kogon, Ebeling, Hauck, **Ruzzo**. “Hardware Acceleration of Short Read Mapping.” *FCCM 2012: 20th IEEE Sym on Field-Programmable Custom Computing Machines*. Toronto, CA, Apr 2012. **(Best Paper Award.)**

9. Briefly list the most recent professional development activities: None

Steve Seitz

1. Education: B.A. Math, Computer Science, UC Berkeley 1991,
 Ph.D. Computer Sciences, Univ. Wisconsin, Madison, 1997
2. Academic experience:
 Professor, Dept. Comp. Sci. and Eng., University of Washington, 2008 -
 Short-Dooley Career Dev. Associate Professor, Dept. Comp. Sci. and Eng.,
 University of Washington, 2005-2008
 Associate Professor, Dept. Comp. Sci. and Eng., University of Washington, 2003-2005
 Assistant Professor, Dept. Comp. Sci. and Eng., University of Washington, 2000-2003
 Adjunct Assistant Professor, Robotics Institute, Carnegie Mellon University, 2000-2005
 Assistant Professor, Robotics Institute, Carnegie Mellon University, 1998-2000
3. Non-academic experience:
 Senior Staff Engineer, Google, 2010-2011 full time, part time 2012 –
4. Certifications or professional registrations: None
5. Current membership in professional organizations: IEEE, ACM
6. Honors and awards:
 IEEE Fellow, 2010
 Short-Dooley Career Development Professorship, 2005
 Alfred P. Sloan Fellowship, 2002
 Office of Naval Research Young Investigator Award, 2001
 David Marr Prize, for the best paper at the 8th International Conference on Computer
 Vision, 2001
 National Science Foundation CAREER Award, 2000
 David Marr Prize, for the best paper at the 7th International Conference on Computer
 Vision, 1999
7. Service activities
 Editorships: International Journal of Computer Vision, 2009-, Editorial Board
 IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001-2006, Associate
 Editor The Visual Computer, International Journal of Computer Graphics, Springer,
 2000-2005, Editorial Board Graphical Models, Academic Press, 2002-2005, Editorial
 Board ACM Computer Graphics Special Issue on Applications of Computer Graphics for
 Computer Vision, November, 1999, guest editor

Selected Program Committees

- Co-chair ICCV (International Conf. on Computer Vision.), 2013
- Co-organizer International Workshop on Video, 2009

Co-organizer BIRS Workshop on Computer Vision and the Internet, 2009
 Co-organizer International Workshop on Computer Vision, 2008
 Area Chair CVPR (IEEE Comp. Vision. Pat. Recog.), 2013, 2008, 2007,
 2004, 2003, 2000
 Area Chair ICCV, 2005, 2003
 Program Committee SIGGRAPH, 2005, 2001, 2000
 Area Chair ECCV (Eur. Conf. on Comp. Vision), 2004
 Program Committee NIPS (Neural Inf. And Proc. Syst.), 2003
 Program Committee AAAI (Amer. Assoc. Art. Intel. Conf.), 2002

UW Committees: exec, undergrad curric, faculty recruiting, space

8. Most important publications in last 5 years:

A. Kushal, B. Self, Y. Furukawa, D. Gallup, C. Hernandez, B. Curless, S. M. Seitz, Photo Tours, *Proc. 3DIMPVT*, 2012, pp. 57-64.

A. Sankar, S. M. Seitz, Capturing indoor scenes with smartphones, *Proc. UIST*, 2012, pp. 403-412

S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz and R. Szeliski, Building Rome in a Day, *Communications of the ACM, Research Highlights*, vol. 54, no. 10, 2011, pp. 105-112.

I. Kemelmacher-Shlizerman, E. Shechtman, R. Garg and S. M. Seitz, Exploring Photobios, *ACM Transactions on Graphics (Proc. SIGGRAPH)*, Aug, 2011.

I. Kemelmacher-Shlizerman, S. M. Seitz, Face reconstruction in the wild, *Proc. Int. Conf. on Computer Vision (ICCV)*, 2011, pp. 1746-1753.

C. Wu, S. Agarwal, B. Curless, S. M. Seitz: Multicore bundle adjustment, *Proc. CVPR*, 2011, pp. 3057-3064.

Y. Furukawa, B. Curless, S. M. Seitz, R. Szeliski. Reconstructing building interiors from images. *Proc. Int. Conf. on Computer Vision (ICCV)*, 2009.

N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal sets for efficient structure from motion. *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2008.

N. Snavely, R. Garg, S. M. Seitz, and R. Szeliski, Finding paths through the world's photos, in *ACM Trans. on Computer Graphics (SIGGRAPH Proceedings)*, vol. 27, no. 3, 2008, pp. 11-21

9. Briefly list the most recent professional development activities: None

Linda G. Shapiro

1. Education:
B.S. Mathematics, University of Illinois, 1970
M.S. Computer Science, University of Iowa, 1972
Ph.D. Computer Science, University of Iowa, 1974
2. Academic experience:
University of Washington, Professor of Computer Science & Engineering, 1990-Present;
Professor of Electrical Engineering, 1989-Present, Associate Professor of Electrical Engineering, 1986-1989, full time.
Virginia Tech, Associate Professor of Computer Science, 1981-1984; Assistant Professor of Computer Science 1979-1981, full time.
Kansas State University, Assistant Professor of Computer Science, 1974-1978, full time.
3. Non-academic experience:
Machine Vision International, Director of Intelligent Systems, Lead researcher for computer vision systems development, 1984-1986, full time.
4. Certifications or professional registrations: none
5. Current membership in professional organizations:
Association for Computing Machinery (ACM); Institute of Electrical and Electronics Engineers (IEEE); Pattern Recognition Society
6. Honors and awards:
Fellow of the IEEE, 1996
Fellow of the IAPR, 2000
Pattern Recognition Society Best Paper Award, 1984, 1989, 1995
MICCAI Workshop on Medical Content-Based Retrieval for Clinical Decision Support Best Paper Award, 2012
7. Service activities (within and outside of the institution)
Editorial Board Member, Pattern Recognition
IEEE PAMI TC Advisory Committee Member
Program Committees: CVPR, MLDM, MICCAI
8. Recent Publications:
R. S. Travillian, K. Diatchka, T. K. Judge, K. Wilamowska, L. G. Shapiro, "An Ontology-Based Comparative Anatomy Information System," *Artificial Intelligence in Medicine*, Vol. 51, Issue 1, 2011, pp. 1-15.

K. Wilamowska, J. Wu, C. Heike, and L. Shapiro, "Shape-Based Classification of 3D Facial Data to Support 22q11.2DS Craniofacial Research," *Journal of Digital Imaging*, Nov 2011.

M. A. Chamblin, R. K. Paasch, D. A. Lytle, A. R. Moldenke, L. G. Shapiro, T. G. Dietterich, "Design of an Automated System for Imaging and Sorting Soil Mesofauna," *Biological Engineering Transactions*, Vol. 4, No. 1, 2012.

I. Atmosukarto, L. G. Shapiro. "3D Object Retrieval Using Salient Views," *International Journal of Multimedia Information Retrieval*, July, 2012

S. Yang, L. Shapiro, M. Cunningham, M. Speltz, C. Birgfeld, I. Atmosukarto, S.-I. Lee, "Skull Retrieval for Craniosynostosis Using Sparse Logistic Regression Models," *Proceedings of the MICCAI Workshop on Medical Content Based Retrieval for Clinical Decision Support*, Springer, 2012. Winner of the Best Paper Award.

B. Soran, Z. Xie, R. Tungaraza, S. Lee, L. Shapiro, T. Grabowski, "Parcellation of Human Inferior Parietal Lobule Based on Diffusion MRI," *IEEE EMBS Conference*, 2012.

R. Tungaraza, J. J. Howbert, S. Mehta, D. Haynor, L. Shapiro, T. Grabowski, "Identifying the Structural Architecture of the Human Inferior Parietal Lobule Using Diffusion MRI," *International Symposium on Biomedical Imaging*, 2012.

V. Raghunath, M. O. Braxton, S. Gagnon, T. Brunye, L. M. reisch, K. A. Allison, D. Weaver, L. G. Shapiro, and J. G. Elmore, "Mouse Cursor Movement and Eye Tracking Data as an Indicator of Pathologists' Attention When Viewing Digital Whole Slide Images," *Journal of Pathology Informatics*, in press. 2013.

R. F. Tungaraza, J. Guan, L. G. Shapiro, J. F. Brinkley, J. Ojemann, and J. D. Franklin, "A Similarity Retrieval Tool for Functional Magnetic Resonance Imaging (fMRI) Statistical Maps," *International Journal of Biomedical Data Mining*, to appear.

9. Briefly list the most recent professional development activities: None for me; I mentor others.

Joshua R. Smith

1. Education
 - Ph.D., Media Arts and Sciences, MIT Media Lab, 1999
2. Academic experience
 - University of Washington, Associate Professor, CSE & EE, 2/2011 - Present
3. Non-academic experience
 - Intel Labs Seattle, Principal Engineer, 7/2004 – 1/2011
 - Tiax LLC, Senior Scientist, 2/2004 – 6/2004
 - Escher Group, Chief Scientist & Director Escher Labs, 11/1998 – 12/2003
4. Certifications or professional registrations: None
5. Current membership in professional organizations: IEEE, ACM
6. Honors and awards
 - **Finalist (one of two) for best paper**, for “Hybrid Analog-Digital Backscatter: A New Approach for Battery-Free Sensing,” IEEE RFID, April 2013.
 - **Best student paper**, for “Hybrid Analog-Digital Backscatter Platform for High Data Rate, Battery-Free Sensing,” WiSNet 2013, January 2013.
 - **Senior Member, IEEE**, January 2013.
 - **Best paper award**, for “An Ultra-Low-Power Human Body Motion Sensor Using Static Electric Field Sensing,” Ubicomp 2012, September 2012.
 - **Nominated for best paper**, for “Optical Localization of Passive UHF RFID Tags with Integrated LEDs” at IEEE RFID 2012, April 2012.
 - **Sezai Innovation Award**, International Society of Rotary Blood Pumps, 2011.
 - **Willem Kolff/Donald B. Olsen Award**, for most promising research in the development of artificial hearts, American Society for Artificial Internal Organs (ASAIO), *June 2011*.
 - **Best Paper**, “A Capacitive Touch Interface for Passive RFID Tags,” *May 2009*, IEEE RFID 2009
 - **Best Demo**, “RFID Sensor Networks with the Intel WISP,” *Nov. 2008*, Sensys 08
 - **Motorola Fellow**, 1995 – 1997, MIT Media Laboratory
 - **Herchel Smith Scholar**, 1991 – 1993, Emmanuel College, University of Cambridge
 - **Phi Beta Kappa**, 1991, Williams College
 - **Sigma Xi**, 1991, Williams College
7. Service activities (within and outside of the institution)
 - IEEE RFID 2013 Technical Program Committee Co-Chair
 - Ubicomp 2013, Technical Program Committee Member
 - NSF Engineering Research Center (ERC) for Sensorimotor Neural Engineering, Thrust Leader for Communications and Interface
 - Intel Science and Technology Center for Pervasive Computing (ISTC-PC), Theme Leader for Low Power Sensing and Communication
8. Selected publications

- **Enabling Seamless Wireless Power Delivery in Dynamic Environments**, A.P. Sample, B.H. Waters, S. Wisdom, J.R. Smith, Proceedings of the IEEE, Vol PP, Issue 99, 2013.
 - **Toward Total Implantability Using Free-Range Resonant Electrical Energy Delivery System: Achieving Untethered Ventricular Assist Device Operation Over Large Distances**, B.H. Waters, A.P. Sample, J.R. Smith, and P. Bonde, Cardiology Clinics, Volume 29, Number 4, pp. 609-625. November 2011.
 - **Analysis, Experimental Results, and Range Adaptation of Magnetically Coupled Resonators for Wireless Power Transfer**, A.P. Sample, D.T. Meyer, J.R. Smith, IEEE Transactions on Industrial Electronics, Vol. 58, No. 2, pp 544-554, Feb 2011.
 - **RFID: From Supply Chains to Sensor Nets**, S. Roy, V. Jandhyala, J.R. Smith, D.J. Wetherall, B.P. Otis, R. Chakraborty, M. Buettner, D.J. Yeager, Y.-C. Ko, A.P. Sample, Proceedings of the IEEE, vol.98, no.9, pp.1583-1592, Sept. 2010.
 - **Design of an RFID-Based Battery-Free Programmable Sensing Platform**. A.P. Sample, D.J. Yeager, P.S. Powledge, A.V. Mamishev, J.R. Smith. IEEE Transactions on Instrumentation and Measurement, Vol. 57, No. 11, Nov. 2008, pp. 2608-2615.
 - **Hybrid Analog-Digital Backscatter Platform for High Data Rate, Battery-Free Sensing**, V. Talla, M. Buettner, D. Wetherall and J.R. Smith. IEEE Topical Conf. Wireless Sensors and Sensor Networks (WiSNet 2013), January 20-23, 2013. **Winner, best student paper award.**
 - **An Ultra-Low-Power Human Body Motion Sensor Using Static Electric Field Sensing**, G. Cohn, T. Lee, S. Gupta, D. Morris, J.R. Smith, M. Reynolds, D. Tan, S. Patel. Ubicomp 2012, Pittsburgh, PA. **Winner, best paper award.**
 - **Optical Localization of Passive UHF RFID Tags with Integrated LEDs**, A.P. Sample, C. Macomber, L.T. Jiang, J.R. Smith, IEEE RFID, April 3-5, 2012. **Nominated for best paper award.**
 - **Promise of unrestricted mobility and freedom with wireless powering of a Ventricular Assist Device (VAD)**, B.H. Waters, A.P. Sample, J.R. Smith, P. Bonde, 19th congress of the International Society of Rotary Blood Pumps, Louisville, KY September 8th to 10th, 2011. **Winner, Sezai Innovation Award.**
 - **Innovative Free-range Resonant Electrical Energy Delivery System (FREE-D system) for a Ventricular Assist Device (VAD) using wireless power**, J.R. Smith, A.P. Sample, B. Waters, Y. Toyoda, R. Kormos, P. Bonde. Am. Soc. for Artificial Internal Organs (ASAIO), Washington, DC, June 10-12, 2011. **Winner, ASAIO Willem J. Kolff / Don B Olsen Award.**
 - **A Capacitive Touch Interface for Passive RFID Tags**, A.P. Sample, D.J. Yeager, J.R. Smith, IEEE RFID, April 27-28, 2009. **Winner, Best Paper award**
 - **RFID Sensor Networks with the Intel WISP**. M. Buettner, B. Greenstein, R. Prasad, A.P. Sample, J.R. Smith, D.J. Yeager, D. Wetherall. 6th ACM Conference on Embedded Networked Sensor Systems (Sensys 2008), November 5-7 2008. **Winner, Best Demo award.**
9. Briefly list the most recent professional development activities
- UW Faculty Fellows teaching development program, Autumn 2010.
 - CELT (Center for Engineering Learning and Teaching) Consultations. Once per quarter 3/2011 – 3/2013

Marty Stepp

1. Education –
M.Sc. Computer Science, University of Arizona, 2003
B.Sc. Computer Science, University of Arizona, 2001
2. Academic experience –
Senior Lecturer, Computer Science & Engineering, U. Washington, 2010–Present.
Lecturer, Computer Science & Engineering, U. Washington, 2006–2010.
Lecturer, Computing and Software Systems, U. Washington Tacoma, 2004–2006.
Lecturer, part time, Computer Science, U. Arizona, 2002-2003.
3. Non-academic experience -
Software Development Engineer, Facebook Corporation, 2012.
Software Development Engineer, Microsoft Corporation, 2003-04.
4. Certifications or professional registrations: None
5. Current membership in professional organizations: ACM, CCSC
6. Honors and awards -
Nominee, UW Community of Innovators Award in Teaching, 2010
Winner, UW CSE Student ACM teaching award, 2009
7. Selected service activities -
Co-chair, committee to revise and reorganize CSE service course curriculum, 2012-13
Committee member for undergraduate admissions and undergraduate curriculum
Assisted with management, training of intro programming undergraduate TA program
Faculty adviser for ACM, ACM-W student organizations, 2007-2009
Judge and faculty sponsor, ACM local programming competition, Autumn 2005-2013
Faculty sponsor, Yahoo! Hack-U programming contest, 2009-2012
8. Selected Publications:
Reges, S. and Stepp, M. *Building Java Programs, 3rd edition*. Pearson Publishing, ISBN 978-0133360905, 2013. An introductory Java programming textbook for CS1/CS2.

Stepp, M., Miller, J., and Kirst, V. *Web Programming Step by Step, 2nd edition*. Lulu Self-Publishing, ISBN 978-1-105-57878-6, 2012. An introductory web programming textbook for students with a CS1 background.

Stepp, M. and Miller, J. A "CS 1.5" *Web Programming Course*. SIGCSE '09 Proceedings of the 40th ACM technical symposium on Computer science education, pp. 121-125. New York, NY, USA: ACM 2009. ISBN 978-1-60558-183-5.

Dan Suciu

1. Education
 - 1991-1995** PhD in Computer and Information Science, **University of Pennsylvania**
 - 1985-1991** M.S. in Mathematics, **University of Bucharest**
 - 1977-1982** M.S. in Computers and Control Engineering, **Polytechnic Institute of Bucharest, Romania**
2. Academic experience
 - 2008 – present** **Full professor**,
University of Washington, Dept. of Computer Science and Engineering.
 - 2002 – 2008** **Associate professor**,
University of Washington, Dept. of Computer Science and Engineering.
 - 2000 – 2002** **Assistant professor**,
University of Washington, Dept. of Computer Science and Engineering.
 - 1998 – present** **Adjunct professor** at the Computer Science Department,
 - 1990 - 1991** **Lecturer**, Polytechnic of Bucharest, Computer Science Department
 - 1984 - 1990** **Assistant Professor**, Polytechnic of Bucharest. Computer Science Department
3. Non-academic experience
 - 2007 – 2008** **Visiting Scientist**, Microsoft Research, Redmond, WA.
 - 1995 - 2000** **Principal member** of the technical staff, AT&T Shannon Laboratories (formerly AT&T Bell Laboratories).
 - 1994 - 1994** **Summer position** AT&T, Bell Laboratories.
 - 1990 - 1991** **Manager** of the Department for Artificial Intelligence, Institute for Computer Research, Bucharest.
 - 1987 - 1990** **Affiliate Researcher**, Institute for Computer Research, Bucharest
 - 1982 - 1984** **Software engineer**, Digital Equipment Maintenance & Service Corporation, Romania.
4. Certifications or professional registrations: 12 US Patents
5. Current membership in professional organizations: ACM member & IEEE member
6. Honors and awards
 - 2013** ICDE 2013 Influential Paper Award, for a paper published in ICDE 2003
 - 2013** ICDT 2013 Best Paper Award
 - 2012** SIGMOD'2012 Best Demonstration Award
 - 2012** ACM PODS Alberto O. Mendelzon Test-of-Time Award, for a paper published in PODS 2002
 - 2011** Elected Fellow of the ACM, 2011
 - 2010** ACM PODS Alberto O. Mendelzon Test-of-Time Award, for a paper published in PODS 2000
 - 2001** NSF Career Award
 - 2001** Alfred P. Sloan Fellow, 2001-2002.
 - 2000** ACM SIGMOD Best Paper Award, for the paper
 - 1996** Morris and Dorothy Rubino_ Award for best doctoral dissertation.

1994 - Doctoral Fellowship from the Institute for Research in Cognitive Science.

1995

1995 ICDT Best Student Paper Award, for the paper

1976 Second Prize at the Eighteens International Mathematic Olympiad, Lienz, Austria.

7. Service activities (within and outside of the institution)

PC Chair: ACM PODS 2007, **PC Co-chair/Vice Chair:** CDT, WWW, ICDE

Conference PC Member: PODS, SIGMOD, ICDE, ICDT, VLDB

Associate Editor: ACM, TWEB, VLDBJ, Information System, ACM TODS

Department Service:

Graduate Program Coordinator, 2011-.

Graduate admissions committee, chair, 2006-2007.

Graduate admissions committee, 2005-2006, 2008-2009, 2009-2010.

Industrial Affiliates Committee, 2000-2001, 2001-2002, 2002-2003, 2003-2004.

Undergraduate CS Admission Committee, 2002.

Faculty mentor for a female science and engineering graduate student (Luna Dong), since 2003.

8. Selected Publications

Refereed Journal Papers

Abhay Kumar Jha, Dan Suciu: *Knowledge Compilation Meets Database Theory: Compiling Queries to Decision Diagrams*. Theory Comput. Syst. 52(3): 403-440 (2013).

Nilesh N. Dalvi, Dan Suciu: *The dichotomy of probabilistic inference for unions of conjunctive queries*. J. ACM 59(6): 30 (2012)

Refereed Conference Papers

Chao Li, Daniel Li, Gerome Miklau and Dan Suciu: *A Theory of Pricing Private Data*. ICDT 2013. **Best paper award**.

Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, Dan Suciu: *Query-based data pricing*. PODS 2012: 167-178. **Invited for submission to a special issue of JACM**.

Alexandra Meliou, Wolfgang Gatterbauer, Katherine F. Moore, Dan Suciu: *The Complexity of Causality and Responsibility for Query Answers and non-Answers*. PVLDB 4(1): 34-45 (2010).

Books

Dan Suciu, Dan Olteanu, Christopher R  , Christoph Koch, *Probabilistic Databases*, Morgan & Claypool Publishers, 2011, ISBN-10 1608456803, ISBN-13 978-1608456802.

Department Colloquium Talks

Query-Based Data Pricing, department colloquium at EPFL, January 2013.

Conference Keynote Talks

An Overview of Probabilistic Databases, 5th LNCC Meeting on Computational Modeling, Petropolis, Brazil, July 2012.

Tractability in Probabilistic Databases, ICDT 2011.

9. Briefly list the most recent professional development activities: None

Steven L. Tanimoto

1. Education

A.B. (Magna Cum Laude), Visual and Environmental Studies, Harvard University, 1971; M.S.E., Electrical Engineering (Image Analysis), Princeton University, 1973; M.A., Electrical Engineering (Image Analysis), Princeton University, 1974; Ph.D., Electrical Engineering (Image Analysis), Princeton University, 1975.

2. Academic experience

University of Connecticut, Assistant Professor (1975-1977).

University of Washington, Assistant Professor (1977-1981).

University of Washington, Associate Professor of Computer Science, and Adjunct Associate Professor of Electrical Engineering (1981-1987).

University of Washington, Professor of Computer Science and Engineering, and Adjunct Professor of Electrical Engineering (1987-present).

3. Non-academic experience

Sylvania Applied Research Laboratory, Electronics Technician, (circuit prototyping and engineering support), 1967-1969: (Summers).

4. Certifications or professional registrations: N/A

5. Current membership in professional organizations

IEEE and IEEE Computer Society; Association for Computing Machinery

6. Honors and awards

IEEE Fellow. International Association for Pattern Recognition Fellow.

2012 International Conference on Pattern Recognition Co-General Chair.

7. Service activities (within and outside of the institution)

IEEE Transactions on Learning Technologies, Steering Committee Chair.

8. Selected publications

A Perspective on the Evolution of Live Programming. (opening keynote talk, and 4- page proceedings paper). Tanimoto, S. L. Proc. Workshop on Live Programming, held in conjunction with the International Conference on Software Engineering, San Francisco, CA. May 19, 2013.

Game Design as a Game. Thompson, R. H. and Tanimoto, S. L. Proc. of the Workshop on Games and Software Engineering, held in conjunction with the International Conference on Software Engineering, San Francisco, CA. May 18, 2013.

CoSolve: A System for Engaging Users in Computer-Supported Collaborative Problem Solving. Fan, S. B., Robison, T., and Tanimoto, S. L., Proc. Int'l. Symp. on Visual Languages and Human-Centric Computing, Innsbruck, pp.205-212, Sept. 2012.

Methods and Systems for Collaborative Formulation and Solution of Problems. Tanimoto, S. L., Fan, S. B., Rice, R. E., Robison, T., and Thompson, R. H. (US Patent and Trademark Office application number 13/679,477, filed Nov. 17, 2012).

An Interdisciplinary Introduction to Image Processing: Pixels, Numbers and Programs. Tanimoto, S. L., Cambridge, MA: MIT Press, 2012.

Picard: A Creative and Social Online Flashcard Learning Game. Tuite, K., Pavlik, T., Fan, S., Robison, T., Jaffe, A., Liu, Y.-E., Andersen, E., and Tanimoto, S. Proc. Foundations of Digital Games, FDG '12, Raleigh, NC, USA., pp. 231-234, May 29-June 1, 2012.

Analyzing a Process of Collaborative Game Design Involving Online Tools. Fan, S. B., Johnson, B. R., Liu, Y.-E., Robison, T., Schmidt, R., and Tanimoto, S. L. Proc. IEEE Symp. On Visual Languages and Human-Centric Computing, Madrid, pp.75-78, 2010.

A game-building environment for research in collaborative design. Tanimoto, S. L., Robison, T., and Fan, S. B. Proc. IEEE Symposium on Computational Intelligence and Games, 2009. CIG 2009, pp.96-103, 2009.

Enhancing state-space tree diagrams for collaborative problem solving. Tanimoto, S. L., Proc. DIAGRAMS 2008, Herrsching, Germany, Sept. 2008.

9. Briefly list the most recent professional development activities
Attended the Workshop on Live Programming, May 2013.
Attended the 2012 International Conference on Pattern Recognition, held in Tsukuba, Japan, November, 2012.
Attended the International Symposium on Visual Languages and Human-Centric Computing, held in Innsbruck, Austria, October 2012.

Ben Taskar

1. Education –
Ph.D. in Computer Science, Stanford University, 2005
M.S. in Computer Science, Stanford University, 2000
B.S. in Computer Science with Distinction, Stanford University, 1998
2. Academic experience –
University of Washington:
Boeing Associate Professor, Computer Science and Engineering, 2013-
University of Pennsylvania
Associate Professor, Computer and Information Science, 2012–2013
Secondary Appointment at Wharton, Department of Statistics, 2011–2013
Magerman Chair Assistant Professor, Computer and Information Science, 2007–2011
3. Non-academic experience -
netDialog, inc., developer, 1998-99, full time.
Workshare, inc., consultant, 2005-2006, part-time
4. Certifications or professional registrations
5. Current membership: ACM
6. Honors and awards:
Boeing Associate Chair (2013-)
Distinguished Research Fellow, Annenberg Center for Public Policy (2011)
NSF CAREER Award (2011-16)
Office of Naval Research Young Investigator Award (2010-13)
Sloan Research Fellowship (2010–12)
Microsoft New Faculty Fellowship Finalist (2010)
DARPA Computer Science Study Group (2009-12)
Magerman Chair Assistant Professorship, University of Pennsylvania (2007–2012) Arthur Samuel Best Thesis Award Runner-up, Stanford University (2005)
Best paper award, Empirical Methods in Natural Language Processing (2004)
Best student paper award, Neural Information Processing Systems (2003)
7. Selected service activities:
Department Service: Co-organizer of Undergraduate Summer Research Program
University Service: Founder and Co-director of PRiML, Penn Research in Machine Learning, a joint SEAS/Wharton center.
Professional Society Service: ACM Dissertation Award Committee, AAA Associate Chair, AAAI Best Paper Award Committee.
Journal Editorial Boards: Action Editor for Journal of Machine Learning Research (JLMR).

Action Editor for Transactions of the Association for Computational Linguistics (TACL).
Editorial Board of Journal of Artificial Intelligence Research (JAIR). Editorial Board of
Journal of Pattern Analysis and Machine Intelligence (PAMI).
Area Chair/Senior Program Committee/Associate Chair/Tutorial Chair:
ICML11, ICML12, ICML13, AISTATS11, NIPS09, NIPS10, UAI08, UAI09, UAI12,
IJCAI09, AAAI08, AAAI13, EMNLP08.

8. Selected Publications:

Determinantal Point Processes for Machine Learning, Alexander Kulesza and Ben Taskar,
Foundations and Trends in Machine Learning: Vol. 5, No 2-3, 2012, 117 pages.

Shape-based Object Detection via Boundary Structure Segmentation, Alexander Toshev, Ben
Taskar, and Kostas Daniilidis, Intl. Journal of Computer Vision, 2011, 24 pages.

Generative-Discriminative Basis Learning for Medical Imaging, Nematollah Batmanghelich,
Ben Taskar and Christos Davatzikos, IEEE Transactions on Medical Imaging Journal, 2011,
19 pages.

Controlling Complexity in Part-of-Speech Induction, Joao Graca, Kuzman Ganchev, Luisa
Coheur, Fernando Pereira and Ben Taskar, Journal of Artificial Intelligence Research, 2011,
24 pages.

Learning from Partial Labels, Timothee Cour, Benjamin Sapp and Ben Taskar, Journal of
Machine Learning Research, 2011, 36 pages.

Posterior Sparsity in Dependency Grammar Induction, Jennifer Gillenwater, Kuzman
Ganchev, Joao Graca, Fernando Pereira and Ben Taskar, Journal of Machine Learning
Research, 2010, 36 pages.

Posterior Regularization in Latent Variable Models, Kuzman Ganchev, Joao Graca, Jennifer
Gillenwater and Ben Taskar, Journal of Machine Learning Research, 2010, 51 pages.

Learning Tractable Word Alignment Models with Complex Constraints, Joao Graca,
Kuzman Ganchev, and Ben Taskar, Journal of Computational Linguistics, 2010, 26 pages.

Joint Covariate Selection and Joint Subspace Selection for Multiple Classification Problems,
Guillaume Obozinski, Ben Taskar, and Michael I. Jordan. Journal of Statistics and
Computing, 2009, 22 pages.

9. Briefly list the most recent professional development activities: None

Emanuil Todorov

1. Education

Doctor of Philosophy: Cognitive Neuroscience, 1998, Massachusetts Institute of Technology

Bachelor of Science: Mathematics, Computer Science, Psychology, 1993, *West Virginia Wesleyan College*

2. Academic experience

Associate Professor, full time, 2009 – present, Applied Mathematics, Computer Science and Engineering; UW

Associate Professor, full time, 2008 – 2009, Cognitive Science; University of California San Diego

Assistant Professor, full time, 2002 – 2008, Cognitive Science; University of California San Diego

3. Non-academic experience: n/a

4. Certifications or professional registrations: n/a

5. Current membership in professional organization

Society for Neuroscience

Society for the Neural Control of Movement

6. Honors and awards

Sloan Research Fellowship in Neuroscience,	2004 – 2006
Howard Hughes Predoctoral Fellowship,	1994 – 1998
Burnett Graduate Fellowship,	1993 – 1994

International Programming Contest of the ACM - 7th Place,	1995
Regional Programming Contest of the ACM - First Place,	1995
Regional Programming Contest of the ACM - Second Place,	1994

International Olympiad in Mathematics - Silver Medal,	1990
International Olympiad in Mathematics - Gold Medal,	1989
International Olympiad in Informatics - Gold Medal,	1989

7. Service activities (within and outside of the institution)

Founder and program chair (with prof. Reza Shadmehr from Johns Hopkins) of the annual symposium series Advances in Computational Motor Control from 2002 to 2011

CSE graduate admissions committee, AMATH chair search committee and mathematical biology curriculum committee

8. Selected publications

Discovery of complex behaviors through contact-invariant optimization
Mordatch I, Todorov E and Popovic, Z (2012). In *ACM SIGGRAPH*

Synthesis and stabilization of complex behaviors through online trajectory optimization
Tassa Y, Erez T and Todorov E (2012). In *IROS*

Infinite-horizon model predictive control for nonlinear periodic tasks with contacts
Erez T, Tassa Y and Todorov E (2011). In *Robotics: Science and Systems*

A unifying framework for linearly-solvable control
Krishnamurthy D and Todorov E (2011). In *Uncertainty in Artificial Intelligence*

A convex, smooth and invertible contact model for trajectory optimization
Todorov E (2011). In *International Conference on Robotics and Automation*

Policy gradients in linearly-solvable MDPs
Todorov E (2010). In *Advances in Neural Information Processing Systems*

Stochastic complementarity for local control of discontinuous dynamics
Tassa Y and Todorov E (2010). In *Robotics: Science and Systems*

Efficient computation of optimal actions
Todorov E (2009). *PNAS*, 106(28): 11478-11483

Structured variability of muscle activations supports the minimal intervention principle of motor control
Valero-Cuevas F, Venkadesan M and Todorov E (2009). *Journal of Neurophysiology*, 102: 59-68

9. Briefly list the most recent professional development activities: none

Martin Tompa

1. Education
1970-74, Harvard, Applied Mathematics, A.B. (magna cum laude)
1974-75, U. Toronto, Computer Science, M.Sc.
1975-78, U. Toronto, Computer Science, Ph.D.
2. Academic experience
1981-84, Assistant Professor, Computer Science, University of Washington.
1984-86, Associate Professor, Computer Science, University of Washington.
1989- , Professor, Computer Science and Engineering, University of Washington.
2001- , Adjunct Professor, Genome Sciences, University of Washington.
2009 , Director, Computational Molecular Biology, University of Washington.
3. Non-academic experience
1985-87, Research Staff Member, Theory of Computation, IBM Research
1987-89, Manager, Theory of Computation, IBM Research
4. Certifications or professional registrations: None
5. Current membership in professional organizations: None
6. Honors and awards
1984-86, Presidential Young Investigator Award, White House Office of Science and Technology Policy
1998, Undergraduate Distinguished Teaching Award, UW chapter of the Association for Computing Machinery
1999, Undergraduate Distinguished Teaching Award, UW chapter of the Association for Computing Machinery
2013, RECOMB 2013 Test of Time Award
7. Service activities (within and outside of the institution)
8. Selected publications

Xiaoyu Chen and Martin Tompa, Comparative assessment of methods for aligning multiple genome sequences. *Nature Biotechnology*, vol. 28, no. 6, June 2010, 567-572.
Huei-Hun Elizabeth Tseng and Martin Tompa, Algorithms for Locating Extremely Conserved Elements in Multiple Sequence Alignments. *BMC Bioinformatics*, vol. 10, no. 432, 2009.
9. Briefly list the most recent professional development activities: None

Daniel S. Weld

1. Education

Ph.D. Artificial Intelligence, MIT, 1988

M.S. Computer Science, MIT, 1984

B.S. Computer Science, B.A. Biophysics and Biochemistry, Yale, 1982

2. Academic experience

University of Washington, Computer Science and Engineering

WRF / T. J. Cable Professor 1999–

Professor 1997-99

Associate Professor 1993-97

Assistant Professor 1988-93

3. Non-Academic Experience

Expert Witness, Google Inc. (2005); Consultant and Venture Partner, Madrona Venture Group (2001-); Founder and Consultant, Asta Networks (2000-2001); Founder and Consultant, Nimble Technology (1999-2001) Sold to Actuate Corporation; Founder and Consultant, AdRelevance Inc. (1998) Sold to MediaMetrix Inc.; Consultant, Excite Inc. (1998); Founder and Consultant, Netbot Inc. (1996-1997); Creator of Jango shopping search; sold to Excite Inc.

4. Certifications or professional registrations

5. Current membership in professional organizations

AAAI (Fellow), ACM (Fellow)

6. Selected Honors

Entrepreneurial Faculty Fellow, University of Washington (2012)

Honorable Mention – Best Student Paper, *17th International World Wide Web Conference (WWW-08)* (2008)

Best Paper, *ACM Conference on Human Factors in Computing Systems (CHI 2008)*

Best Paper, *16th ACM Conf. on Information and Knowledge Management (CIKM 2007)*

WRF / T. J. Cable Endowed Professorship, University of Washington (1999–)

7. Selected Service Activities

Member, NSF CISE AC Subcommittee on Industry (2008-2009); Member, AAAI Publications Access Committee (2007-2008); Member, RIACS Science Council (2005-2008) Member, International Joint Semantic Web Services Consortium (SWSC) Architecture Subcommittee (2003).

Area Editor, *Journal of the ACM* (2007-2012); Editorial Board Member, *Artificial Intelligence* (1999-2008).

IUI Program Chair (2009); IUI Associate Chair (2007); AAAI (1996), Senior-Member Paper Track (2006), Challenge Track (2010).

Program Committee Member for over 25 conferences.

Reviewer for 10 journals, NSF.

8. **Selected Recent Publications**

Dai, P., Lin C., Mausam, and Wed, D. "POMDP-Based Control of Workflows for Crowdsourcing," *Artificial Intelligence* To appear 2013.

L.B. Chilton, G. Little, D. Edge, D.S. Weld, J.A. Landay, "Cascade: Crowdsourcing Taxonomy Creation," (CHI-13), April 2013.

Kolobov, A., Mausam, Weld, D. "Discovering Hidden Structure in Factored MDPs," *Artificial Intelligence*, **189**:19-47, September 2012.

D.S. Weld, E. Adar, L. Chilton, R. Hoffmann, E. Horvitz, M. Koch, J. Landay, C. Lin, and Mausam "Personalized Online Education — A Crowdsourcing Challenge" 4th Human Computation Workshop (HCOMP-12), July 2012.

Lin, C., Mausam, and Weld, D. "Crowdsourcing: Dynamically Switching between Synergistic Workflows," (AAAI-12), July, 2012.

Zhang, C., Hoffmann, R., and Weld, D. "Ontological Smoothing for Relation Extraction with Minimal Supervision," (AAAI-12), July, 2012.

Ling, X, and Weld, D. "Fine Grained Entity Resolution," (AAAI-12), July, 2012.

A. Kolobov, Mausam, and D. Weld, "A Theory of Goal-Oriented MDPs with Dead Ends" Conference on Uncertainty in Artificial Intelligence (UAI), 2012.

C.H. Lin, Mausam, and D.S. Weld, "Crowdsourcing Control: Moving Beyond Multiple Choice" Conference on Uncertainty in Artificial Intelligence (UAI), 2012.

Kobolov, A, Mausam, and Weld, D. "Reverse Iterative Deepening for Finite Horizon MDPs with Large Branching Factors," (ICAPS-12). June, 2012.

Amershi, S., Fogarty, J. and Wed, D. "[ReGroup: Interactive Machine Learning for On-Demand Group Creation in Social Networks](#) (*CHI-12*), May 2012.

Dai, P., Mausam, Weld, D. and Goldsmith, J. "Topological Value Iteration Algorithms" *Journal of AI Research*, **42**, September, 2011.

Gajos, K., Wobbrock, J. and Weld, D., "Automatically Generating Personalized User Interfaces with SUPPLE" *Artificial Intelligence*, **174**(12-13):910-950, August, 2010.

9. Briefly list the most recent professional development activities: none

David Wetherall

1. Education
Ph.D., Computer Science, Massachusetts Institute of Technology, 1998
E.E., Electrical Engineering, Massachusetts Institute of Technology, 1995
S.M, Computer Science, Massachusetts Institute of Technology, 1994
B.E., Electrical Engineering, University of Western Australia, 1989
2. Academic Experience
University of Washington, Professor (CSE), 2011-2013, full time
University of Washington, Associate Professor (CSE), 2004-2011, full time
University of Washington, Assistant Professor (CSE), 1999-2004, full time
3. Non-Academic Experience
Intel, Director of Intel Research Seattle, ran research lab, 2006-2009, full time
WATRI, Visitor, research collaboration (sabbatical), 2006-2007, full time
Asta Networks, Co-founder and Chief Architect, 2000-2000, full time
QPSX Communications, Member of Technical Staff, 1989-1991
4. Certifications or professional registrations: None
5. Current membership in professional organizations: ACM (SIGCOMM, SIGMOBILE, SIGOPS), IEEE (Communications Society), USENIX.
6. Honors and awards
Fellow, IEEE, 2013
Fellow, ACM, 2012
ACM MobiSys, Best Paper Award, 2008
ACM SIGCOMM, Test-of-Time Award, 2007
IEEE Communications Society Bennett Prize, 2005
Sloan Fellow 2004
NSF Career Award 2002
ACM SIGCOMM Best Student Paper Award, 2002
7. Service activities
2012: SIGCOMM PC member, SIGCOMM CCR Area Editor
2011: Mobisys PC Co-chair, SIGCOMM workshop PC Co-chair (2 workshops), IMC PC member, SIGCOMM CCR Area Editor, CSE Executive Committee
2010: Mobisys PC member, SIGCOMM workshop PC member, SIGCOMM CCR Area Editor, CSE Executive Committee
2009: Mobicom PC member, NSDI PC member, UW/MSR Institute organizer
2008: SIGCOMM General Co-chair and PC member, NSDI PC member

8. Selected publications

- “Ambient Backscatter: Communication Out of Thin Air,” Liu, Parks, Talla, Gollakota, Wetherall and Smith, *ACM SIGCOMM*, 2013.
- “Expressive Privacy Control with Pseudonyms,” Han, Lui, Pu, Peter, Krishnamurthy, Anderson and Wetherall, *ACM SIGCOMM*, 2013.
- “Demystifying Page Load Performance with Wprof,” Wang, Balasubramanian, Krishnamurthy and Wetherall, *USENIX NSDI*, 2013
- “Detecting and Defending Against Third-Party Tracking on the Web,” Roesner, Kohno, Wetherall, *USENIX NSDI*, 2012.
- Advancing the State of Home Networking,” Edwards, Grinter, Mahajan and Wetherall, *Communications of the ACM*, 2011.
- “These Aren’t the Droids You’re Looking For,” Hornyack, Han, Jung, Schechter, Wetherall, *ACM CCS*, 2011.
- “Augmenting data center networks with multi-gigabit wireless links,” Halperin, Kandula, Padhye, Bahl, Wetherall *ACM SIGCOMM*, 2011.
- “Odessa: Enabling Interactive Perception Applications on Mobile Devices,” Ra, Sheth, Mummert, Pillai, Wetherall, Govindan, *ACM MobiSys*, 2011.
- “A Software Radio-based UHF RFID Reader for PHY/MAC Experimentation,” Buettner and Wetherall, *IEEE RFID*, 2011.
- “Dewdrop: An Energy-Aware Runtime for Computational RFID,” Buettner, Greenstein and Wetherall, *USENIX NSDI*, 2011.
- “RFID: From Supply Chains to Sensor Networks,” Roy, Jandhyala, Smith, Wetherall, Chakraborty, Buettner, Yeager, You-Chang, and Sample, *Proc. of the IEEE*, 2010.
- “Computer Networks,” 5th edition, U.S. and international plus translations, Tanenbaum and Wetherall, *Pearson Education*, 2010.
- “Predictable 802.11 Packet Delivery from Wireless Channel Measurements,” Halperin, Wu, Sheth and Wetherall, *ACM SIGCOMM*, 2010.
- “When I am on Wi-Fi, I am Fearless’,” Klasnja, Consolvo, Jung, Greenstein, LeGrand, Powledge, Wetherall, *ACM CHI*, 2009.
- “TVA: A Dos-limiting Network Architecture,” Yang, Wetherall and Anderson, *ACM/IEEE Trans. on Networking*, 2008.
- “Privacy Oracle,” Jung, Sheth, Greenstein, Wetherall, Maganis, Kohno, *ACM CCS*, 2008.
- “Taking the Sting out of Carrier Sense: Interference cancellation for wireless LANs,” Halperin, Anderson and Wetherall, *ACM MOBICOM*, 2008.
- “An Empirical Study of UHF RFID Performance,” Buettner and Wetherall, *ACM MOBICOM*, 2008.
- “Improving Wireless Privacy with an Identifier-Free Link Layer Protocol,” Greenstein, McCoy, Pang, Kohno, Seshan and Wetherall, *ACM MobiSys*, 2008.
- “Studying Black Holes in the Internet with Hubble,” Katz-Bassett, Madhyastha, John, Krishnamurthy, Wetherall, Anderson, *USENIX NSDI*, 2008.
- “Reducing Network Energy Consumption via Sleeping and Rate-Adaptation,” Nedeveschi, Popa, Iannaccone, Ratnasamy, Wetherall, *USENIX NSDI*, 2008.

“Passport: Secure and Adoptable Source Authentication,” Lui, Li, Yang, and Wetherall, *USENIX NSDI*, 2008.

“Wirelessly-Charged UHF Tags for Sensor Data Collection,” Yeager, Prasad, Wetherall, Powledge and Smith, *IEEE RFID*, 2008.

9. Professional development activities:

Visitor at Telefonica studying cellular networks (sabbatical), 2012-2013.

MOOC Instructor and Developer (Coursera) for “Computer Networks”, 2013.

John Zahorjan

1. Education
PhD Computer Science, University of Toronto, 1980
M.Sc. Computer Science, University of Toronto, 1976
B.Sc. Applied Mathematics, Brown University, 1975
2. Academic experience
University of Washington, Department of Computer Science & Engineering
Professor 1989-
Associate Professor 1985-1989
Assistant Professor 1980-1985

University of Paris VI, Laboratoire MASI
Professeur Associe 1987-1988, Winter 1992
3. Non-academic experience
Quantitative System Performance, Inc.
Principal 1982-2006
4. Certifications or professional registrations
5. Current membership:
6. Honors and awards:
2010 ACM SIGMETRICS Test of Time Award
Presidential Young Investigator Award, 1984-1990
7. Selected service activities:
Program Committee Member, IPTPS 2008 (7th Intl. Workshop on Peer to Peer Systems).
Technical Program Committee Member, WinMee/WiTMemo 2007 (3rd Intl. Workshop on Wireless Network Measurement / 3rd Intl. Workshop on Wireless Traffic Measurements and Modeling).
Technical Program Committee Member, PerCom 2006 (4th Annual IEEE International Conference on Pervasive Computing and Communications).
Technical Program Committee Member, PerCom 2005 (3rd Annual IEEE International Conference on Pervasive Computing and Communications).
Program Committee Member, 18th ACM Symposium on Operating Systems Principles (2001).
Department Graduate Admissions Chair, 2007-09

8. Selected Publications:

Aruna Balasubramanian, Ratul Mahajan, Arun Venkataramani, Brian Levine, and John Zahorjan. "Interactive WIFI Connectivity for Moving Vehicles", Proceedings of ACM SIGCOMM, August 2008.

Ratul Mahajan, John Zahorjan, and Brian Zill, "Understanding Wi-Fi Based Connectivity From Moving Vehicles", Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference, October 2007.

Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan, "Analyzing the MAC-level Behavior of Wireless Networks in the Wild", Proceedings of ACM SIGCOMM, September 2006.

Charlie Reiss, Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan, "Measurement-Based Models of Delivery and Interference in Static Wireless Networks", Proceedings of ACM SIGCOMM, September 2006.

Thomas Anderson, Andrew Collins, Arvind Krishnamurthy, and John Zahorjan, "PCP: Efficient Endpoint Congestion Control," 3rd Symposium on Networked Systems Design & Implementation (NSDI), May 2006.

Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan, "Sustaining Cooperation in Wireless Networks." 2nd Symposium on Networked Systems Design & Implementation (NSDI), May 2005.

9. Briefly list the most recent professional development activities

Luke Zettlemoyer

10. Education

B.S., Computer Science and Applied Mathematics, North Carolina State University, 2000
Ph.D., Computer Science, Massachusetts Institute of Technology, 2009

11. Academic experience

University of Washington, Assistant Professor, 2010-present, full time

12. Non-academic experience: n/a

13. Certifications or professional registrations

14. Current membership in professional organizations

Association for the Advancement of Artificial Intelligence
Association for computational Linguistics

15. Honors and awards

NSF CAREER Award, 2013
DARPA Computer Science Study Group, 2011
NSF International Research Fellowship, 2009
Best Paper Award: ACL 2009
Best Paper Award: UAI 2005
Graduate Research Fellowships: NSF, NDSEG, and Microsoft, 2000-2006

16. Service activities (within and outside of the institution)

Co-organizer: Machine Learning Journal Special Issue on Learning Semantics, 2012;
AAAI Workshop on Grounded Language Physical Systems, 2012; Pacific Northwest
Symposium on Natural Language Processing, 2012

Program Committee Area Chair: EMNLP 2011, NAACL 2012, ICML 2013, EMNLP
2013

Regular Program Committee / Reviewer: AAAI, AISTATS, ACL, COLING, CoNLL,
EMNLP, ICML, NIPS, UAI, Artificial Intelligence Journal, Journal of Artificial
Intelligence Research, IEEE Transactions on Robotics

17. Selected Publications

- a. Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. A Joint Model of Language and Perception for Grounded Attribute Learning. To appear in *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.
- b. Yoav Artzi and Luke Zettlemoyer. Bootstrapping Semantic Parsers from Conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2010: 421-432.

- c. Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Inducing Probabilistic CCG Grammars from Logical Form with Higher-order Unification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP), 2010: 1223-1233.
- d. S.R.K. Branavan, Harr Chen, Luke S. Zettlemoyer, and Regina Barzilay. Reinforcement Learning for Mapping Instructions to Actions. In *Proceedings of the Joint Conference of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing* (IJCNLP-ACL) 2009: 82–90. [*Best Paper Award*]
- e. Luke S. Zettlemoyer and Michael Collins. Learning Context-dependent Mappings from Sentences to Logical Form. In *Proceedings of the Joint Conference of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing* (IJCNLP-ACL) 2009: 976–984.

18. Briefly list the most recent professional development activities

Appendix C – Equipment

Student Computing Resources (physically accessible)

General Purpose Instructional Laboratories

- 77 Dell Workstations
- Eighty 24" LCD monitors
- Thirty 20" LCD monitors

Animation Instructional Laboratory

- 22 Dell Workstations
- Sixteen 24" LCD monitors
- Eighteen 20" LCD monitors

Hardware Laboratory

- 52 Dell Workstations
- Twenty 20" LCD monitors
- Forty Four 17" LCD monitors
- 18 Tektronix Logic Analyzers

Collaboration devices

- 2 SMART Boards

Printers

- 7 HP LaserJet Mono Printers

Instructional Staff Computing

- 50 Faculty Desktops
- 150 Graduate Student Desktops/Laptops (acting in a TA role)

Instructional Shared Facilities Hosted in Datacenter

- Instructional Cycle Cluster (provides remote access to course software)
 - 4 Dell PowerEdge 2950s, dual quad-core 2.8GHz processors, 32GB RAM
- Instructional File Server (student home directories):

- Dell PowerEdge 2950, 16GB RAM, 18TB Storage Capacity
- Instructional Web Server (course webs, course projects, course archives)
 - Dell PowerEdge R520, 32GB RAM, 18TB Storage Capacity
- Instructional Virtualization Infrastructure (on demand course project servers, VDI)
 - 6 Dell PowerEdge 710s, dual 6-core 2.4Ghz processors, 24GB RAM
 - 1 Dell PowerEdge R310 with attached PowerVault MD3200i
- storage array

Animation Renderfarm

- 18 PowerEdge servers, 2.4Ghz , 8GB RAM

Appendix D – Institutional Summary

1. *The Institution*

- a. Name and address of the institution:
University of Washington, Seattle, WA 98195
- b. Chief executive officer of the institution:
Michael K. Young, President
- c. Self-study report submitted by:
Judy Ramey, Interim Dean, College of Engineering

Effective July 15, 2013, Michael B. Bragg will be the new Dean of the College of Engineering.

- d. Institution (University of Washington) is accredited by:
Northwest Commission on Colleges and Universities (NWCCU)

Initial University of Washington accreditation evaluation:
April 1918

Most recent University of Washington accreditation evaluation:
Accreditation was reaffirmed based on the Spring 2011 Year One Evaluation.

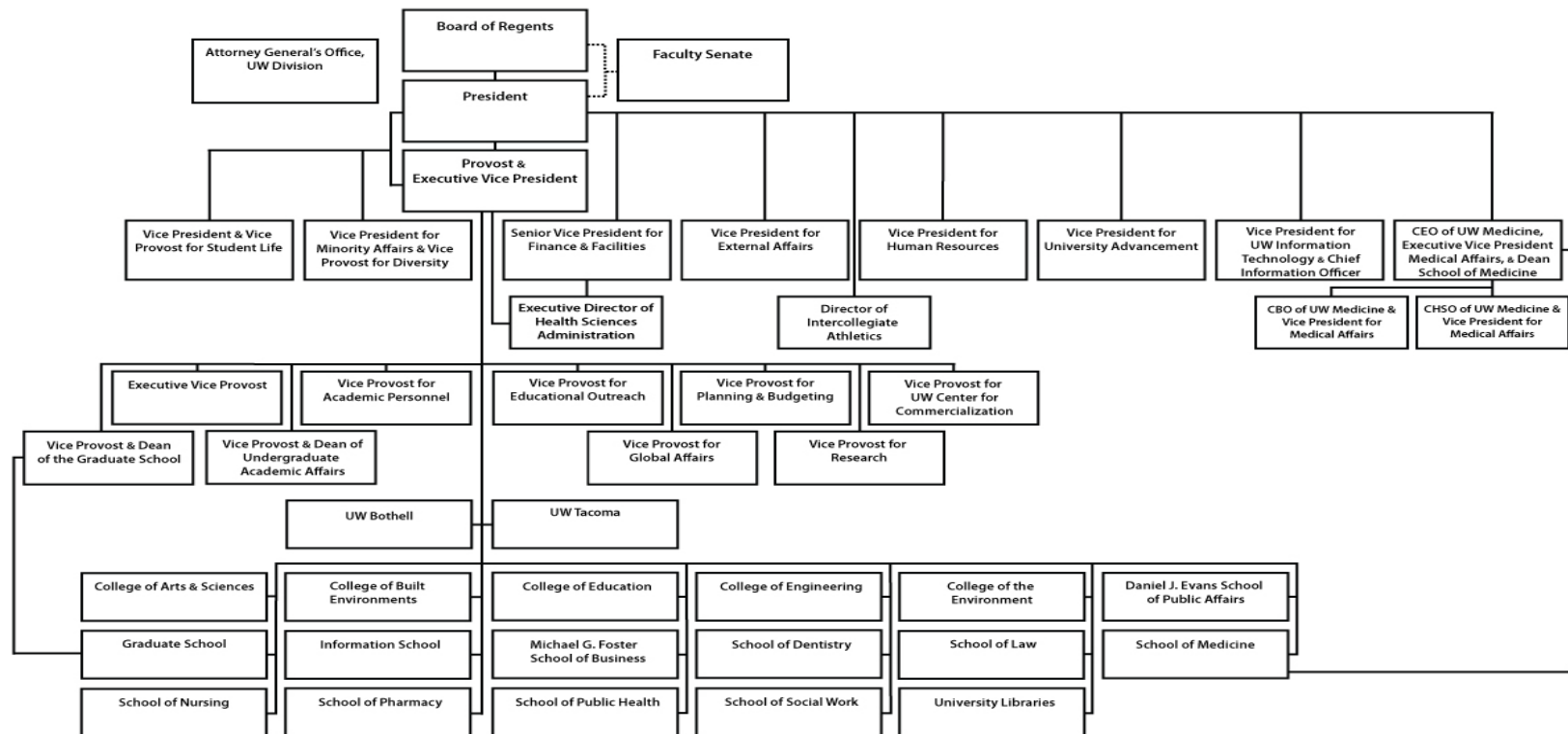
2. *Type of Control*

State-assisted Public Research University

3.. *Educational Unit*

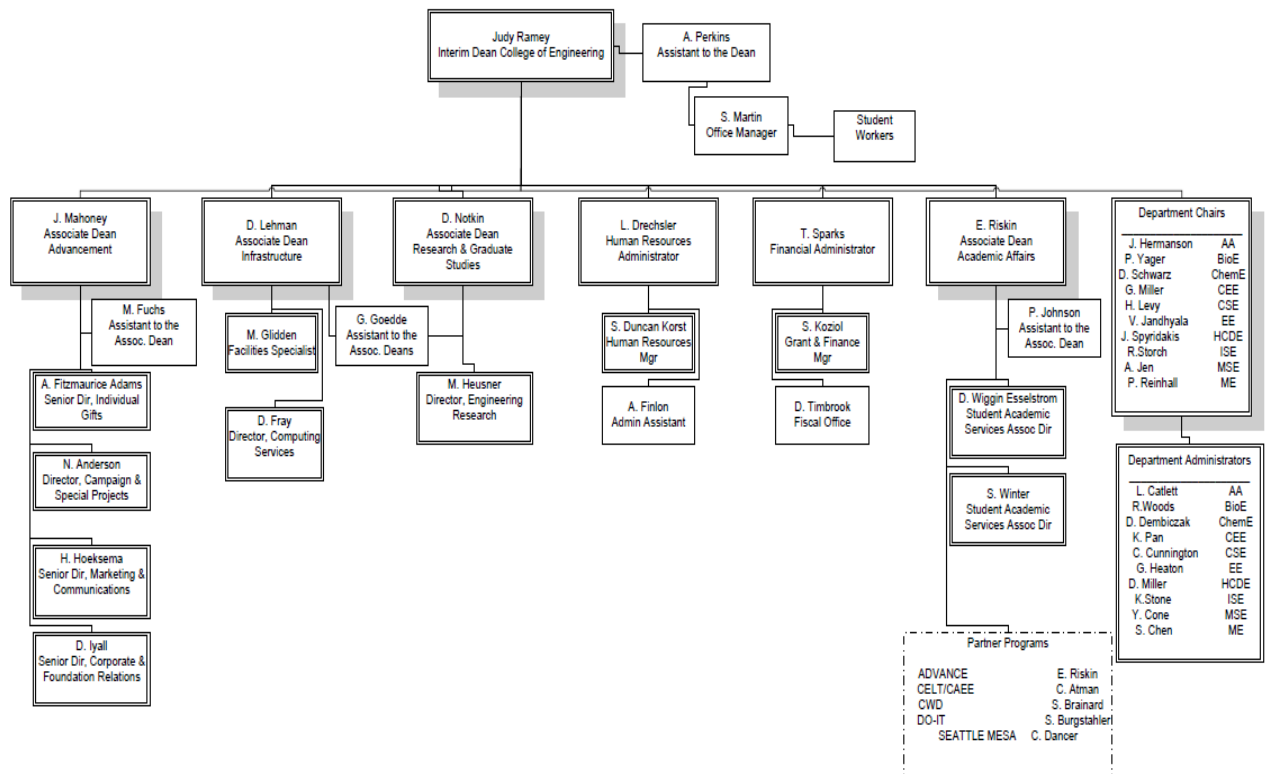
The University Organization Chart shows the position of the College of Engineering within the University of Washington. The College of Engineering is a separately organized unit with its own budgetary and program control within the University of Washington. Judy Ramey became the Interim Dean of the College effective January 1, 2013, when the previous Dean of the College, Matthew O'Donnell, stepped down. Judy Ramey reports to the Provost and Executive Vice President, Ana Mari Cauce.

University Organization Chart (Approved by the President by authority of the *Board of Regents Governance*, Standing Orders, Chapter 1) This chart reflects the reporting relationships of the University of Washington's administrative offices, schools, colleges, and campuses. Select any box on this chart to link to APS 1.2, University Wide Leadership List, where more information is available.



The College of Engineering Organizational Chart and the College's listing in the University Wide Leadership List show the engineering academic departments / programs and their reporting relationship within the College of Engineering. The names of the College of Engineering Department Chairs are listed as well as names and titles of the administrative heads of the principal units of the College of Engineering.

COLLEGE OF ENGINEERING ORGANIZATIONAL CHART



University Wide Leadership List

Engineering, College of

Acting Dean—Judith A. Ramey
Assistant to the Dean—Andrea Perkins

Academic Affairs

Associate Dean—Eve A. Riskin

Computing Services

Director—David T. Fray

Development and External Relations	Assistant Dean—Judy K. Mahoney
Infrastructure	Associate Dean—Dawn Lehman
Research and Graduate Studies	Associate Dean—David Notkin
Departments	
Aeronautics and Astronautics	Chair—James Hermanson
Bioengineering (with School of Medicine)	Chair—Paul Yager
Chemical Engineering	Chair—Dan Schwartz
Civil and Environmental Engineering	Chair—Greg Miller
Computer Science and Engineering	Chair—Henry (Hank) Levy
Electrical Engineering	Chair—Vikram Jandhyala
Human-Centered Design and Engineering	Chair—Jan Spyridakis
Industrial and Systems Engineering	Chair—Richard Storch
Materials Science and Engineering	Chair—Alex Jen
Mechanical Engineering	Chair—Per Reinhall
Engineering Programs	
Disabilities Opportunities Internetworking Technology (DO-IT)	Director—Sheryl Burgstahler
Mathematics, Engineering, Science Achievement (MESA)	Director—James Dorsey
Workforce Development, Center for (CWD)	Director—Suzanne G. Brainard

3. 1. University of Washington Engineering Initiative

Objective

Send nearly 2,000 additional Engineers into the Washington workforce over the next 10 years by leveraging partnerships, requiring relatively modest additional State investment.

Washington State: “Innovation is in our nature”

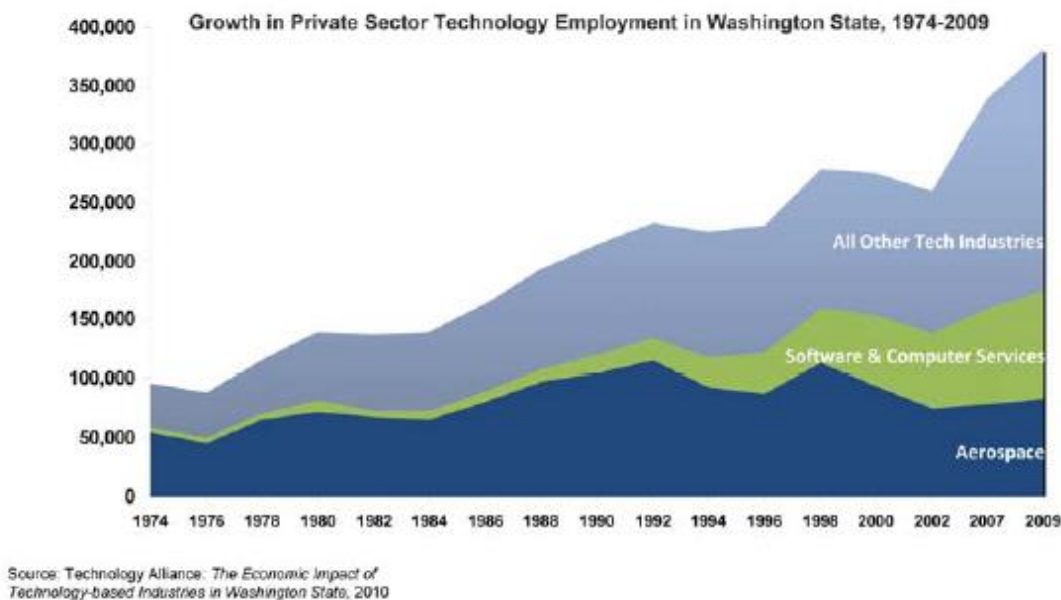
Innovation defines Washington State. In the most recent Kauffman Foundation *New Economy Index*, Washington ranks second overall among the 50 states, behind only Massachusetts. As shown in Figure 1, technology industry employment in Washington

has quadrupled since 1974.

STEM graduates drive innovation

Individuals with degrees in STEM fields (Science, Technology, Engineering, and Mathematics) drive this innovation. They create and staff the companies. And these jobs create other jobs: economists estimate that the 381,000 jobs in Washington's technology industries support nearly 827,000 other jobs throughout our state economy, accounting for a total of more than 1.2 million jobs – equivalent to 42% of Washington employment.

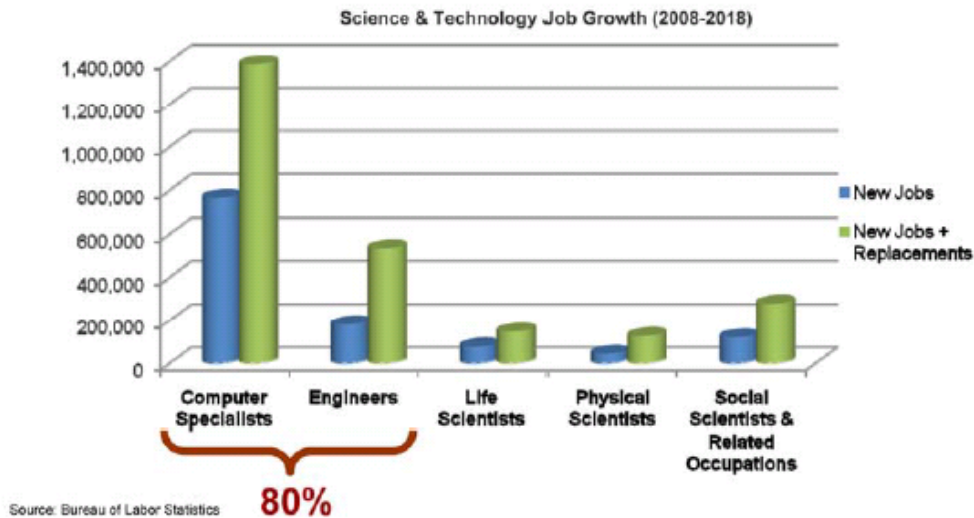
Figure 1: Washington's technology industry employment has quadrupled since 1974.



Not all STEM is created equal

When it comes to future job growth, “not all STEM is created equal.” As shown in Figure 2, the U.S. Bureau of Labor Statistics projects that in the current decade, 80% of all new jobs in all STEM fields will be in Computer Science and other fields of Engineering. Washington State workforce projections mirror this.

Figure 2: STEM (Science, Technology, Engineering, and Mathematics) careers are where the growth is – but *not all STEM is created equal*.



Washington STEM degree production does not match Washington workforce needs

Unfortunately, Computer Science and other fields of Engineering are precisely where Washington's higher education system has the greatest shortfall. Washington ranks 45th among the 50 states (and 7th among ten top technology states) in Computer Science degree production relative to the number of jobs in the field (Figure 3). Washington similarly ranks 45th (and 9th among ten top technology states) in Engineering degree production relative to the number of jobs in the field (Figure 4). Washington also ranks among the bottom states in *per capita* degree production in these fields, at both undergraduate and graduate levels.

Figure 3: Washington ranks very low in Computer Science degree production relative to Computer Science occupations.

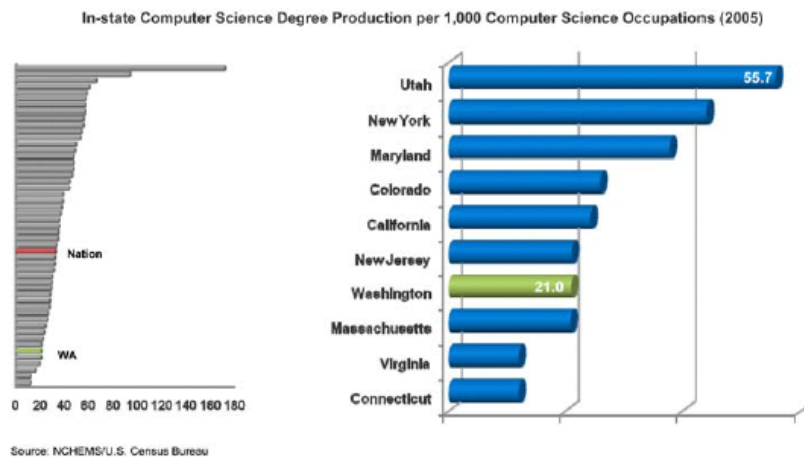
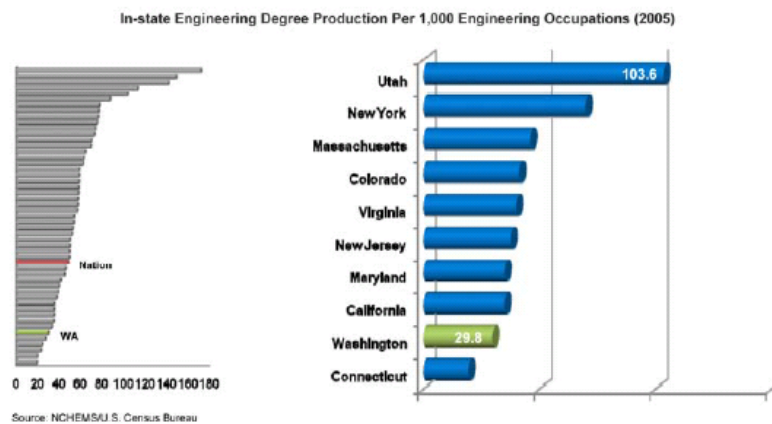


Figure 4: Similarly for other fields of Engineering.



A recent analysis by the Washington State Higher Education Coordinating Board indicates that Computer Science and other fields of Engineering rank first and second, among all fields requiring a bachelors education or greater, in the gap between workforce demand and degree production (Figure 5). A recent analysis of WorkSource job postings and Occupational Employment Statistics data by the *Seattle Times* shows that Computer Science and related fields dominate *all* job availability in Washington (Figure 6). This gap between supply and demand in Computer Science and other fields of Engineering threatens the future of Washington's economy, and it deprives our children of the opportunity to be first-tier participants in the innovation economy.

Figure 5: In Washington, the gap between demand and supply is greater in Computer Science and other fields of Engineering than in all other fields.

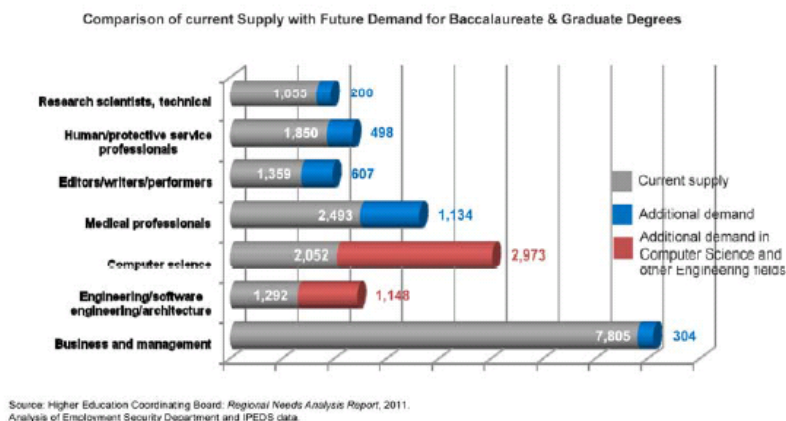
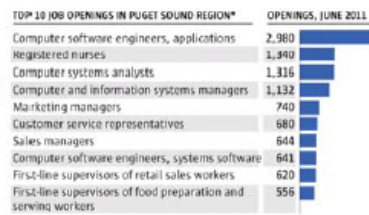


Figure 6: Currently, Computer Science dominates all job availability in Washington.

57% of the job openings among the top 10 occupations are in computing.

Where the jobs are and aren't

Some employers are hiring, but the openings don't overlap much with the jobs most commonly lost to the economic downturn.



* King, Snohomish, Pierce and Kitsap counties



Sources: Seattle Times analysis of WorkSource job postings and Occupational Employment Statistics data

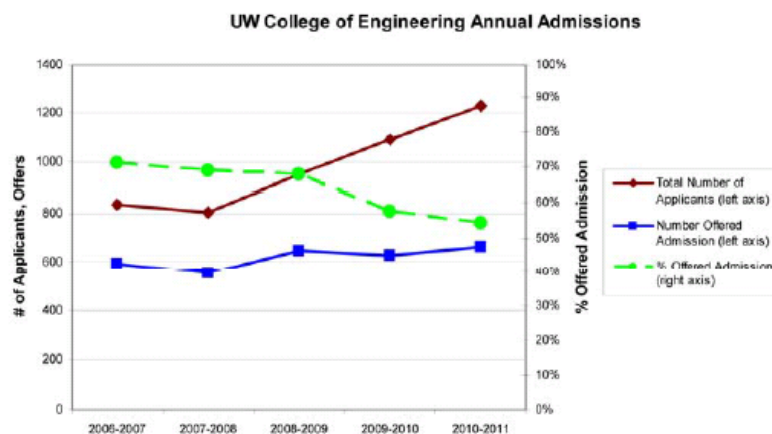
MARK NOWLIN / THE SEATTLE TIMES

Source: Used with permission from the Seattle Times.

At UW, we have the student demand – we lack the program capacity

Although nationally there is a pipeline issue, at the University of Washington the limitation is program capacity, not student interest. As shown in Figure 7, in the most recent year, the UW College of Engineering was able to accommodate only 54% of undergraduate applicants. More than 500 undergraduates seeking to major in a UW Engineering program – *students who were already enrolled at UW or at a Washington State Community and Technical College and who had successfully fulfilled the pre-requisites for entry to Engineering* – had to be turned away last year. In the Department

Figure 7: It's a capacity problem! Last year the UW College of Engineering had to turn away more than 500 prospective undergraduate majors.



of Computer Science & Engineering, the most over-subscribed program in the College of Engineering, 70% of applicants had to be turned away – only 30% could be accommodated. More than 40% of the students that the College of Engineering was unable to accommodate, and more than 60% of the students that the Department of Computer Science & Engineering was unable to accommodate, had college grade point averages of 3.25 or above – successful college students are being turned away due to lack of capacity.

The University of Washington Engineering Initiative

Responding to the needs of Washington State's economy, technology businesses, and students, the University of Washington has crafted an opportunity for 425 additional students to join the College of Engineering.

This initiative will require a financial partnership between the University, UW students, and the State, including \$3.75 million per year in new State funding. In the past, the State funding request for an Engineering enrollment increase of this size would have been roughly \$8.5 million – 2.25 times as great. Under the University of Washington Engineering Initiative, however:

- Overall University of Washington enrollment will not increase as a result of the Initiative. Rather, responding to student demand, UW will shift enrollments towards Engineering from other fields. Thus, only the “marginal cost” of an Engineering education must be funded – the “base cost” of each student's education is already covered.
- A substantial proportion of the tuition generated by Engineering students will be retained by the College of Engineering, decreasing the subsidy required. (Engineering students take roughly half their courses in other fields – the remaining tuition helps to defray these costs.)
- The State will fund only the remaining “gap” – the marginal increment required to produce an Engineering degree, minus the portion covered by tuition.

Impact and scope of the initiative, however, also relies on the state's ability to mitigate additional budget reductions to higher education.

The University of Washington Engineering Initiative is an extremely cost-effective investment in the future of our state, our technology businesses, and our children. As a result of this initiative, supported by a relatively modest, targeted investment on the part of the State, the University of Washington will be able to send nearly 190 additional engineering graduates annually into the workforce – almost 2,000 over the next decade.

4. Academic Support Units

Each Engineering department requires certain courses in mathematics, natural sciences, communication, and engineering fundamentals.

Unit

Applied Mathematics

Head

Jose Nathan Kutz, Professor and Chair

Biology	H.D. ‘Toby’ Bradshaw, Professor and Chair
Chemistry	Paul B. Hopkins, Professor and Chair
Mathematics	Selim Tuncel, Professor and Chair
Physics	Blayne R. Heckel, Professor and Chair
Statistics	Elizabeth Thompson, Professor and Chair

5. Non-academic Support Units

<u>Unit</u>	<u>Head</u>
UW Enrollment Services	Philip Ballinger, Assistant Vice President for Enrollment, Admissions
UW Career Center	Susan Terry, Director
UW Information Technology (UW-IT)	Kelli Trosvig, Vice President and CIO
University of Washington Libraries	Lizabeth (Betsy) A. Wilson, Dean
College of Engineering Office of Student Academic Affairs Academic Services	Eve Riskin, Associate Dean for
UW ADVANCE Center for Institutional Change (CIC)	Eve Riskin, Professor and Director
Center for Engineering Learning & Director Teaching (CELT)	Cynthia J. Atman, Professor and
Center for Workforce Development Director	Suzanne Gage Brainard, Executive
College Computing Services	David T. Fray, Director
Disabilities, Opportunities, Internetworking, and Technology (DO-IT)	Sheryl Burgstahler, Director

Education at a Distance for Growth and Excellence (UW/EDGE) and Engineering Professional Programs (EPP)

David P. Szatmary, Vice Provost
UW Educational Outreach

Engineering Library

Mel DeSart, Head

GenOM – Genomics Outreach for Minorities Project

Lisa Peterson, Director

Seattle MESA (Mathematics, Engineering, Science Achievement

Clarence Dancer, Director

6. Credit Unit

The College of Engineering adheres to the traditional ratio of one contact hour and two outside hours per week for each credit of coursework, which is the University guideline. Contact hours can include many different formats, including laboratories and quiz sections. As a rule of thumb, two hours of scheduled lab or quiz per week counts toward 1 credit, but exceptions may be warranted. Considering the flexibility allowed by the University and the diversity of teaching styles and learning environments, the College of Engineering's Council on Educational Policy will consider approving courses that do not meet these guidelines. For such courses, faculty should present written material justifying the departure from the traditional ratio and, additionally, should be prepared to justify the course credit/contact hour ratio to the Council on Educational Policy in person.

7. Tables

Table D-1. Program Enrollment and Degree Data

CE - computer engineering
CS - computer science

Academic Year		Year				Total Enrollment	Degrees Awarded
		1st	2nd	3rd	4th		
2012-2013	CE	20	62	48	42	172	52
	CS	38	130	119	130	417	151
2011-2012	CE	34	59	44	49	186	55
	CS	48	130	142	111	431	122
2010-2011	CE	33	37	65	43	178	51

	CS	63	151	126	111	451	128
2009-2010	CE	29	63	53	35	180	39
	CS	68	161	119	107	455	134
2008-2009	CE	28	48	49	45	170	53
	CS	81	129	110	96	416	93

Table D-2 Personnel

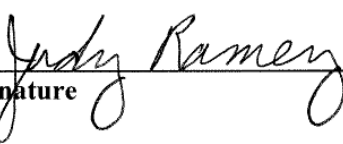
Table D-2. Personnel							
COMPUTER ENGINEERING							
Year ¹ : Autumn 2012 (Active Employees)							
		HEAD COUNT		FTE			
		FT	PT				
Administrative ²		18	5	20.63			
Faculty (tenure-track) ³		47	4	38.68			
Other Faculty (excluding student Assistants)		42	106	41.10			
Student Teaching Assistants ⁴		30	105	32.00			
Technicians/Specialists		47	12	44.50			
Office/Clerical Employees		7	19	8.40			
Others ⁴ Student Research Assistants		131	76	131.75			
College of Engineering notes:							
· TA and RA at 50% or greater equals 1 FTE							
· Data source is the CoE Data Resources website.							
ABET notes:							
Report data for the program being evaluated.							
¹ Data on this table should be for the fall term immediately preceding the visit. Updated tables for the fall term when the ABET team is visiting are to be prepared and presented to the team when they arrive.							
² Persons holding joint administrative/faculty positions or other combined assignments should be allocated to each category according to the fraction of the appointment assigned to that category.							
³ For faculty members, 1 FTE equals what your institution defines as a full-time load.							
⁴ For student teaching assistants, 1 FTE equals 20 hours per week of work (or service). For undergraduate and graduate students, 1 FTE equals 15 semester credit-hours (or 24 quarter credit-hours) per term of institutional course work, meaning all courses — science, humanities and social sciences, etc.							
⁴ Specify any other category considered appropriate, or leave blank.							

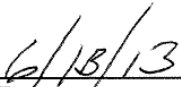
Signature Attesting to Compliance

By signing below, I attest to the following:

That Computer Engineering has conducted an honest assessment of compliance and has provided a complete and accurate disclosure of timely information regarding compliance with ABET's *Criteria for Accrediting Engineering Programs* to include the General Criteria and any applicable Program Criteria, and the ABET *Accreditation Policy and Procedure Manual*.

Judy Ramey, Interim Dean


Signature


Date