

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

January 2019

Table of Contents

Contents

Table of Contents	2
BACKGROUND INFORMATION	7
A. Contact Information	7
B. Program History	7
C. Options	8
D. Program Delivery Modes	8
E. Program Locations.....	9
F. Public Disclosures	10
G. Deficiencies, Weaknesses or Concerns from Previous Evaluation(s) and the Actions Taken to Address Them	10
CRITERION 1. STUDENTS.....	11
A. Student Admissions.....	11
B. Evaluating Student Performance	13
C. Transfer Students and Transfer Courses	14
C.1 Acceptance of Transfer Students.....	14
C.2 Evaluation of Transfer Credit	15
D. Advising and Career Guidance	15
D.1 Advising	15
D.2 Career Guidance	16
E. Work in Lieu of Courses	17
F. Graduation Requirements	18
G. Transcripts of Recent Students	18
CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES	19
A. Mission Statement.....	19
Mission Statement—UW	19
B. Program Educational Objectives	19
C. Consistency of the Program Educational Objectives with the Mission of the Institution....	20
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
Table 4-1 Information sources for assessment.....	24
Table 4-2 Assessment procedures for student outcomes	25
A.2 Assessment Process Overview	26
Figure 4-1 Summary of Department Assessment and Improvement Processes	27
A.3 Targeted Assessment of Outcomes in Courses	29
A.4 Additional Assessment Processes	34

A.4.1 Employer and Alumni feedback	34
A.4.2 Co-op feedback	34
A.4.3 End-of-program interviews	37
A.4.4 Student surveys upon exit from the department.....	38
A.4.5 Student course evaluations	40
A.4.6 Student Anonymous Feedback.....	41
A.4.7 TA Review	42
A.4.8 Faculty self-evaluations	42
A.4.9 Capstone design projects.....	42
A.4.10 Curriculum Committee	43
B. Continuous Improvement.....	43
B.1 Targeted Assessment of Outcomes in Courses	43
B.2 Additional Assessment Processes	44
Table 4-6 Courses Introduced Since 2013	45
Table 4-7 Additional Changes	48
CRITERION 5. CURRICULUM.....	50
A. Program Curriculum.....	50
A.1 Tables	50
Table 5-1 Curriculum.....	51
A.2 Curriculum Alignment with Program Educational Objectives	56
A.3 Curriculum & Student Outcomes	57
Table 5-2 Student outcomes and mapping to curricular and extra-curricular elements. ...	57
Table 5-3 Student outcomes and mapping to curriculum requirements	58
A.4 Course Prerequisite Structure.....	60
A.5 and A.6 Meeting Degree Requirements	62
A.7 Design Experience	65
A.8 Extra-Curricular Activities and Co-op	73
A.8.1 Extra-Curricular Activities.....	73
A.8.2 Co-op Credits	73
A.9 Materials Available to Committee	73
B. Course Syllabi	73
CRITERION 6. FACULTY.....	74
A. Faculty Qualifications	74
Table 6-1 Faculty Qualifications.....	76
B. Faculty Workload.....	82
Table 6-2 Faculty Workload Summary.....	82
C. Faculty Size	87
D. Professional Development	87
E. Authority and Responsibility of Faculty	87
CRITERION 7. FACILITIES	88
A. Offices, Classrooms and Laboratories	88
B. Computing Resources	90
C. Guidance.....	91
D. Maintenance and Upgrading of Facilities	91

E. Library Services.....	91
F. Overall Comments on Facilities	92
CRITERION 8. INSTITUTIONAL SUPPORT	93
A. Leadership.....	93
B. Program Budget and Financial Support	93
C. Staffing.....	94
D. Faculty Hiring and Retention.....	94
E. Support of Faculty Professional Development.....	96
PROGRAM CRITERIA	100
APPENDICES	101
Appendix A – Course Syllabi	101
Written & Oral Communication	101
ENGR 231 – Introduction to Technical Communications.....	101
Mathematics & Natural Sciences.....	103
CHEM 142 General Chemistry.....	103
CHEM 152 General Chemistry.....	105
MATH 124 Calculus with Analytic Geometry I.....	107
MATH 125 Calculus with Analytic Geometry II	108
MATH 126 Calculus with Analytic Geometry III.....	109
MATH 307 Introduction to Differential Equations	111
MATH 308 Matrix Algebra with Applications	112
PHYS 121 Mechanics	114
PHYS 122 Electromagnetism	115
PHYS 123: Waves	116
STAT/MATH 390: Probability and Statistics in Engineering and Science.....	118
Required.....	122
CSE 142 Computer Programming	122
CSE 143 Computer Programming II.....	124
CSE 332 Data Structures	131
EE 205 Introduction to Signal Conditioning	137
EE 371: Design of Digital Circuits and Systems	139
CE System Electives	142
CSE 401 Introduction to Compiler Construction.....	143
CSE 402 Design and Implementation of Domain-Specific Languages.....	145
CSE 403 Software Engineering	147
CSE 428 Computational Biology Capstone.....	149
CSE 441 Advanced HCI: Advanced User Interface Design, Prototyping, and Evaluation	151
CSE 444 Database Systems Internals	153
CSE 451 Introduction to Operating Systems	155
CSE 452 Introduction to Distributed Systems	157
CSE 460 Animation Capstone	159

CSE 461 Introduction to Computer-Communication Networks.....	161
CSE 469 Computer Architecture I.....	163
CSE 470 / EE 470 Computer Architecture II.....	165
CSE 474 Introduction to Embedded Systems	167
CSE 475 Embedded Systems Capstone	169
CSE 481A Capstone Software Design: Operating Systems	172
CSE 481C Capstone Software Design: Robotics.....	174
CSE 481N NLP.....	176
CSE 481S Security.....	178
CSE 481V AR/VR	180
Appendix B – Faculty Vitae	182
Anderson, Richard	182
Anderson, Ruth E.....	184
Anderson, Thomas E.....	186
Balazinska , Magdalena	191
Beame, Paul	193
Bricker, Lauren	195
Ceze, Luis.....	198
Choi, Yejin	200
Curless, Brian L.	204
Domingos, Pedro.....	206
Farhadi, Ali	208
Fogarty, James	210
Fox, Dieter	213
Froehlich, Jon E.	215
Gollakota, Shyamnath.....	217
Grossman, Dan.....	218
Heimerl, Kurtis.....	222
Hemingway, Bruce.....	224
Jamieson, Kevin	225
Karlin, Anna R	226
Krishnamurthy, Arvind	228
Kemelmacher-Shlizerman, Ira (Irena)	230
Kohno, Tadayoshi	232
Lin, Huijia (Rachel)	233
Lee, James R.	235
Mones, Barbara	239
Oskin, Mark	240
Patel, Shwetak N.	242
Perkins, Hal.....	244
Rao , Anup	245
Reges, Stuart	246
Reinecke, Katharina	248
Roesner, Franziska	250
Ruzzo, Walter L.	253

Schafer, Hunter	256
Seelig, Georg	257
Seitz, Steve	259
Shapiro, Linda G.	261
Smith, Joshua R.	263
Suciu, Dan	265
Tanimoto, Steven L.	267
Tatlock, Zachary	269
Taylor, Michael Bedford	272
Tompa, Martin	274
Wang, Xi	275
Weld, Daniel S.	277
Zahorjan, John	281
Appendix C – Equipment	283
Appendix D – Institutional Summary	285
1. The Institution	285
2. Type of Control	285
3. Educational Unit	285
4. Academic Support Units	287
5. Non-academic Support Units	287
6. Credit Unit	288
7. Tables	288
Table D-1. Program Enrollment and Degree Data	288
Table D-2. Personnel	290
Submission Attesting to Compliance	291

BACKGROUND INFORMATION

A. Contact Information

List name, mailing address, telephone number, fax number, and e-mail address for the primary pre-visit contact person for the program.

Professor Arvind Krishnamurthy
Paul G. Allen School of Computer Science & Engineering
Box 352350
University of Washington
Seattle WA 98195-2350

Email: arvind@cs.washington.edu
Phone: (206) 616-0957

B. Program History

Include the year implemented and the date of the last general review. Summarize major program changes with an emphasis on changes occurring since the last general review.

The University of Washington's Department of Computer Science & Engineering began as an inter-college graduate program in 1967. In 1975 a bachelor's degree 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 degree program -- an ABET-accredited Bachelor of Science in Computer Engineering degree 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. In 2013 we expanded again up to 250 bachelor's degrees. In 2015 we grew to 345 degrees, and in 2017 to 370. In 2017 - our 50th anniversary year - the University of Washington Board of Regents voted to create the Paul G. Allen School of Computer Science & Engineering, elevating the status of CSE within the university and linking us in perpetuity with the internationally renowned investor, philanthropist and computing pioneer.

CSE currently has roughly 70 faculty, 70 technical and administrative staff members, 500 graduate students (350 in the full-time program and 150 in the Professional Master's Program), and 1,300 undergraduate students. We will soon award more than 450 bachelor's degrees per year of which about 90 will be Computer Engineering.

As of 2019, the School continues to offer a **Bachelor of Science in Computer Engineering** degree through the College of Engineering and a **Bachelor of Science in Computer Science** through the College of Arts & Sciences. While in the past we did not set quotas between Computer Science and Computer Engineering, we now have a target of maintaining approximately 20% of our degrees for Computer Engineering. 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. We also offer a combined bachelor's/master's program for Computer Engineering and Computer Science students. Admissions is competitive; admitted students can complete a master's within 3 or 4 quarters of their bachelor's degree.

Our last ABET review was in 2013. We have added many new courses in the last 6 years, and in 2015 went through a significant Computer Engineering curriculum update. The changes we made are described in detail in Criterion 5 of this document. Summarizing, the changes include:

- Added 300 and 400 level courses
- Cross listed courses in Electrical Engineering and Computer Engineering.
- Configured Engineering Electives to more easily ensure that students meet the ABET engineering requirements.
- Increased the number and diversity of capstone courses.

C. Options

List and describe any options, tracks, concentrations, etc. included in the program.

We do not currently offer any options within our Computer Engineering degree.

D. Program Delivery Modes

Describe the delivery modes used by this program, e.g., days, evenings, weekends, cooperative education, traditional lecture/laboratory, off-campus, distance education, web-based, etc.

Almost all undergraduate students in the department are enrolled full-time on-campus, although there are some part-time students. Courses are offered during day-time hours all four quarters of the academic year, although many fewer courses are offered during summer quarter. 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, extenuating circumstances like medical leaves, and for students nearing graduation with minimal outstanding courses to complete.

Historically, most students coming out of high school have had to first be admitted to the university, and then after some time here to our major. Typically, students applied to enter our program in their sophomore year at UW. In 1999 we introduced a "Direct Admission" program, something new to us and to our university. Under that program, a small number of high-achieving applicants were admitted directly into our major out of high school. By removing the uncertainty about admission to the department, we hoped to improve our success rate in attracting these students. This worked extremely well for us and for the university. About a quarter of our majors have come through our Direct Admission program.

Starting in the Fall of 2018, we (and the rest of the College) moved to a Direct to College (DTC) model for Computer Engineering. We have since disbanded our departmental Direct Admission program for Computer Engineering and transitioned to this new, college-level process. While we haven't yet enrolled a cohort of DTC students, we are excited by the prospects and encouraged by the continued interest in Computer Engineering. In the future, we expect about half our

Computer Engineering majors to be direct admits, about 16% to be transfer students, and about 34% to be students admitted after approximately one year at the UW. Further details on DTC are provided in the Admission section.

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 (UW's mascot is the Husky/Dawg). Our extensive diversity and outreach programs are discussed in the Criterion 1.

E. Program Locations

Include all locations where the program or a portion of the program is regularly offered (this would also include dual degrees, international partnerships, etc.).

Our program takes place on the University of Washington - Seattle campus. Our School is housed in two state-of-the-art buildings: the Paul G. Allen Center for Computer Science & Engineering and the recently completed Bill and Melinda Gates Center for Computer Science & Engineering. Classrooms for our courses tend to be near these two buildings, although sometimes they are spread throughout the UW campus.

We also offer four international direct exchange agreements:

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

All four 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.

Our longest agreement is with KTH in Stockholm, Sweden, which began in Autumn 2001. We sent our first students to ETH in Spring 2007, to Saarland in 2010, and our newest agreement is with EPFL. We signed our agreement with EPFL in Spring 2015, with the first students heading abroad for the 2015-16 school year.

These programs continue to thrive, with many applicants each year for each available position abroad. We plan to continue these programs in the future and are working on ways to continue expanding our study abroad options to ensure that as many students as possible have access to high quality computer science and engineering content while also becoming well-rounded global citizens.

F. Public Disclosures

Provide information concerning all the places where the Program Education Objectives (PEOs), Student Outcomes (SOs), annual student enrollment and graduation data is posted or made accessible to the public. If this information is posted to the Web, please provide the URLs.

Program URL: <https://www.cs.washington.edu/ABET>.

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

Summarize the Deficiencies, Weaknesses, or Concerns remaining from the most recent ABET Final Statement. Describe the actions taken to address them, including effective dates of actions, if applicable. If this is an initial accreditation, it should be so indicated.

The only concern expressed from our ABET 2013 review was that students might be able to graduate without completing the required 45 credits of math and science and 67.5 Engineering units. There was concern that as changes were made to the program, students could fail to meet these requirements. To address this possibility, we have now added an explicit component to the degree that requires all students to complete additional engineering courses. This is checked automatically by the university's Degree Audit Reporting System (DARS).

GENERAL CRITERIA

CRITERION 1. STUDENTS

A. Student Admissions

Summarize the requirements and process for accepting new students into the program.

Until Fall 2018, UW students generally applied to majors at the end of their sophomore year – a situation encouraged by Washington state’s higher education policy that anticipates a large number of transfers from our two-year community colleges. Unusually, our department was also admitting roughly 20% of our students through freshman Direct Admission. Direct Admission was a better experience for the students as they could start off as Computer Engineers from day one without waiting one or two years uncertain if they would eventually be admitted. While we’ve expanded this program for our Computer Science degree, for our Computer Engineering students, we have transitioned to a new, college-wide Direct to College (DTC) freshman pathway. This past year, Engineering programs adopted a policy of admitting up to 50% of future graduating classes directly as freshman as part of DTC. These students enter the College on admission to the University. They are then placed into Engineering programs between the end of their freshman year and the middle of their sophomore years. This change in admissions has a very positive impact on degree programs; with the additional certainty and time in the major this program affords students, they can better plan their major requirements and are more likely to consider and undertake research and cooperative education experiences that enhance their education. We still admit roughly 34% of our students from the current UW student body. These are generally students who came to UW intending to study other majors but are excited by Engineering and apply to our program after finishing a series of prerequisite courses. We also admit about 16% of our students directly as transfer students from other colleges, primarily Washington state’s two-year schools. Finally, we also take small number of students through the College of Engineering [STARS program](#): STARS is a two-year program with a specialized curriculum designed to build learning skills and strengthen academic preparation for core math and science prerequisites. STARS scholars are guaranteed placement into an engineering or computer science major.

Our Undergraduate Advising Office coordinates the departmental undergraduate admissions effort. Admission to the department is highly competitive. As mentioned above, there are four paths into the major (Direct to College placement, STARS students, current UW student admission, and transfer student admission). All paths except STARS, 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 pathway for current UW students, 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), 5 credits of science (Physics 121) and 5 credits of English Composition. Our Admissions Committee is comprised of faculty members and undergraduate advisors.

All 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 notifications about three weeks after the application deadline. There are two standard admissions cycles per year, with deadlines of July 1 for Autumn and January 15th for Spring.

Our School is very active in recruiting the best students to our program. We host information sessions several times 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 Robinson School 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.

Our efforts supporting diversity and inclusion have expanded greatly over the past few years, with new activities, better engagement from Allen School students and faculty, and more attention to supporting underrepresented minority and low-income students (along with continued programming for women in computing). Comprehensive lists of ongoing and recent work can be found at:

- <https://www.cs.washington.edu/diversity/ongoing-activities>
- <https://www.cs.washington.edu/diversity/latest-initiatives>

Current efforts include:

- We expanded our K-12 outreach. A team of paid student Ambassadors (majority underrepresented) reaches more than 1,500 K-12 students annually through workshops, schools visits, and tours. Summer camps host 200+ students, half girls, with scholarships available for low-income students. We now participate in UW pre-college programs such as the College of Engineering Math Academy and initiatives through the Office of Minority Affairs & Diversity. And we continue to lead teacher workshops and participate in an annual NCWIT Award for Aspirations in computing.
- We instituted a holistic admissions process utilizing non-academic factors to promote diversity.
- New academic support courses for Intro Programming to serve low-income engineering and computer science students in the College of Engineering STARS program.
<https://www.engr.washington.edu/stars>
- A new seminar for our diverse population of transfer students provides community, academic advice, and connection to resources.
- A new one-month summer class for incoming freshmen from underrepresented backgrounds has served 70 students: half women, plus URM students, low-income students, and students with disabilities. The course teaches CS skills along with basic college preparation and community development.
<https://www.cs.washington.edu/students/ugrad/directadmission/startup>

- We helped our students create two new organizations: a Student Advisory Council focused on fostering a supportive undergrad community, and a Q++ organization for LGBTQ+ students.
- A new Diversity Committee of faculty, staff, and students supports people working on diversity and equity initiatives through feedback, help with projects, and connections to resources and leadership. <https://www.cs.washington.edu/diversity/committee>
- We published Best Practices for Inclusive Teaching in CS: https://docs.google.com/document/d/15egU3IYmgd8c2exue3G7oqpoSQmOVYRumu_Nq7PnkCQ/edit

The impact of these efforts is conveyed in Figure 1.1, which shows the fraction of female students in our first introductory programming course). The overall percent of women enrolled in our undergraduate program is now around 30% (20% for Computer Engineering alone).

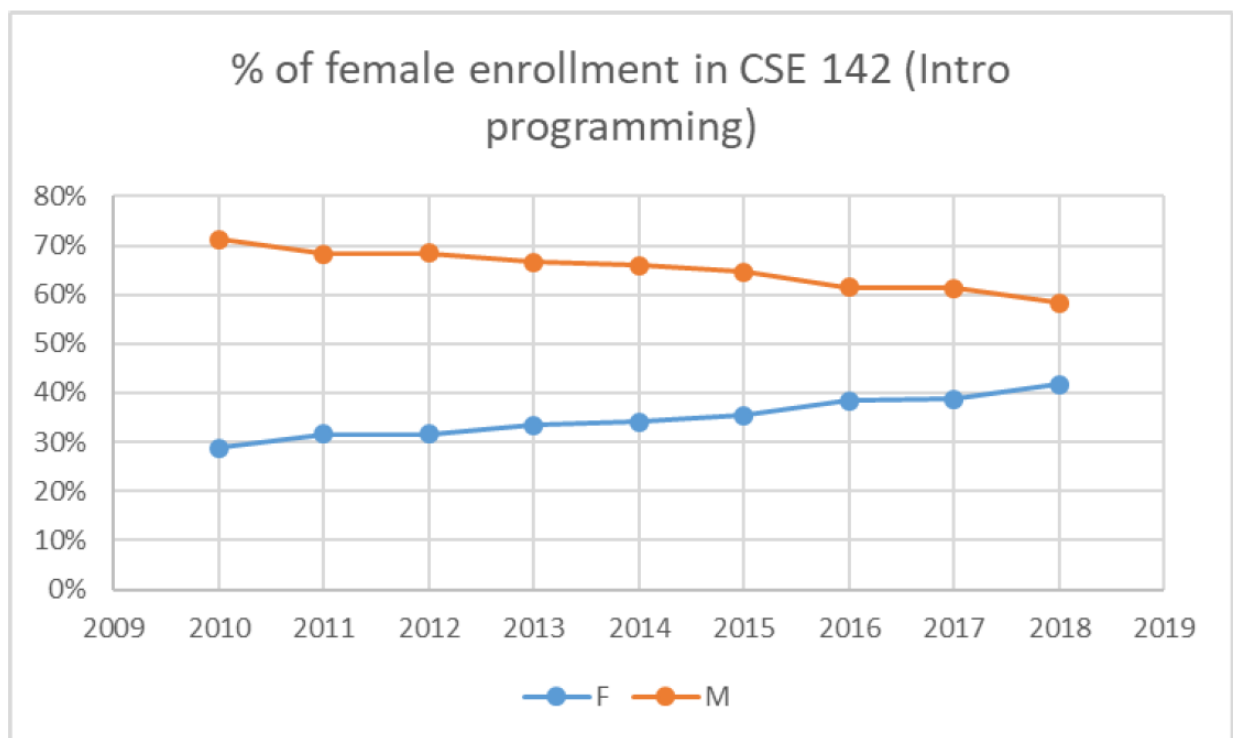


Figure 1.1

B. Evaluating Student Performance

Summarize the process by which student performance is evaluated and student progress is monitored. Include information on how the program ensures and documents that students are meeting prerequisites and how it handles the situation when a prerequisite has not been met.

Students are assigned a decimal grade between 0.0 and 4.0 for each course they complete at UW. Instructors are responsible for grades, with new instructors receiving both broad written guidelines and one-on-one mentoring.

There are no formal guidelines for the assignment of grades and our faculty do not grade on a curve. However, we do aim for consistency across offerings of courses and across our curriculum. Toward that end, we have developed rough guidelines for assignment of course grades that we share with instructors, with a clear indication that these are guidelines not prescriptions and that an individual course offering might deviate from the guidelines for a number of reasons. Per the guidelines, the average course grade is typically 3.3-3.5, with less than 10% of students receiving a 4.0 and most students receiving a 3.0 or higher. Students must receive at least a 2.0 to obtain degree credit, or else file a petition with the Undergraduate Faculty Program Coordinator.

Our satisfactory progress policy requires that students must successfully complete three courses toward graduation every non-summer quarter. The advising staff reviews student progress every quarter. This allows them to quickly identify struggling students and to connect them with tutoring and other campus resources that might help struggling students 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. Instructors are contacted each quarter by the advising staff seeking early information on any students particularly struggling or particularly striving.

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 courses the student took at another institution, although occasionally they are based on prior work experience. Requests for prerequisite waivers are handled with an online form and then 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.

C. Transfer Students and Transfer Courses

Summarize the requirements and process for accepting transfer students and transfer credit. Include any state-mandated articulation requirements that impact the program.

C.1 Acceptance of Transfer Students

Transfer students first apply to the University of Washington, and then they are evaluated for the departmental admissions process. There is significant interest by transfer students in our major, and our advising staff meets with many of them during their visits 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 the State of Washington community colleges may petition to have 135 transfer credits applied to their degree. However, 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.

Community college course work generally covers only our introductory programming courses (CSE 142 and 143), calculus, science, and English Composition. 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. If the courses completed are on our list of pre-approved transfer courses (primarily, from the state’s community colleges) then credit is granted immediately. Otherwise, the student is asked to go through the petition process described below in the Evaluation of Transfer Credit section. The possible outcomes are 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. The overwhelming majority of our transfer credits are from Washington State Community Colleges with which we have course articulation agreements.

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, etc. 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 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.

D. Advising and Career Guidance

Summarize the process for advising and providing career guidance to students. Include information on how often students are advised, who provides the advising (program faculty, departmental, college or university advisor).

D.1 Advising

We have an Undergraduate Advising Office with eight full-time professional staff members: Crystal Eney, Director of Student Services; Raven Avery, Assistant Director of Diversity and Outreach; Jenifer Hiigli, Lead Academic Adviser; Maggie Ryan, Senior Academic Adviser; Chole Dolese, Senior Academic Adviser; Leslie Ikeda, Academic Adviser; and Kim Nguyen, Career Advising Specialist. We also have a Course Coordinator (Pim Lustig) and an Outreach Assistant (Jeremy Munroe). The Faculty Undergraduate Program Coordinator (Arvind Krishnamurthy) supervises the overall advising effort. Our departmental advising office provides both pre-major advising for prospective students and advising and curriculum planning for current majors. The CE program is well documented and can be found online at:

- For current students: <https://www.cs.washington.edu/academics/ugrad>, which summarizes in detail the program requirements and advising process.
- For prospective students, <https://www.cs.washington.edu/academics/ugrad/admissions>, 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 master’s (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 serves both 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.

Visits to advising are optional, but the majority of our students (over 70% of all CE majors) visit advising every year. In the 2019-2020 academic year we will be initiating a primary adviser model where one adviser will be responsible for tracking a smaller group of students. This will further ensure that all students have an additional layer of support.

D.2 Career Guidance

Since our last ABET review in 2013, we have significantly grown our career guidance resources. Most importantly, this past year we hired Kim Nguyen to our advising team as a Career Advising Specialist. Kim is an alumna of UW who earned Computer Engineering and Electrical Engineering degrees and then worked at Microsoft as a program manager and finally as a company recruiter. She has ramped up the career guidance and preparation services we provide our students. In addition to now being able to provide one-on-one counseling appointments, we offer group presentations and small group activities to get students prepared for industry. This fall (2019), we are introducing a 1 credit seminar on “everything you need to know on how to land a job in the software industry”. Over the last couple of years, our students had expressed a desire for stronger career preparation through various feedback channels like surveys, lunch with the director, and the like. Kim has led our response. Below we document the career events that are now available:

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 Google, Indeed, Code.org, and Salesforce.

Internship Panel (Winter Quarter)

The internship panel helps students interested in internships. The panel consists of recent CSE grads from companies such as Amazon, Zillow, Google and Microsoft. The panelists provide advice on how to secure the best-fitting internships, how to prepare for those internships, and what to expect during the internship. 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 Quarters)

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 Quarters)

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. The interviews 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 and Winter Quarters)

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.

Recruiting 101 (Fall, Winter and Spring Quarters)

This presentation covers introductory information about what students can expect and how they should prepare for their internship or job hunt. Topics covered include an organized strategy to approaching the job search, how to effectively use time at the career fair, basics of a resume, how to correspond with recruiters, recruiting timelines, and what to expect in a technical interview.

Negotiating Your First Job Offer (Fall, Winter and Spring Quarters)

This presentation covers basic strategies students can use to negotiate full-time offers. We cover how to evaluate the monetary aspects of a job offer, including stock, stock options, relocation, salary, sign-on bonuses, yearly bonuses, and health benefits. Discussion of more qualitative aspects of an offer, like location, job content, career trajectory, and work/life balance, helps students understand how to holistically evaluate their options. Current industry trends for compensation and benefits are discussed so that proper expectations can be set. Several scenarios are walked through to show examples of how to navigate various negotiation situations.

E. Work in Lieu of Courses

Summarize the requirements and process for awarding credit for work in lieu of courses. This could include such things as life experience, Advanced Placement, dual enrollment, test out, military experience, etc.

Except for a limited number of co-op credits (ENGR 321 and CSE 301), 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 70% of our undergraduates participate in co-ops or internships. If a student signs up for internship credit under the CSE 301 or ENGR 321 course heading, they receive 2 credits for a 3 month internship. They are required to submit several documents/assignments and surveys to earn the credit. While a student may accrue up to 6 total credits of CSE 301 as general credits towards their degree, they can use at most 2, CSE 301 credits, as electives toward the Computer Engineering degree requirements.

[Advanced Placement policies](#) are set by the University of Washington. Generally speaking, appropriate credit is given for AP exam scores of 3, 4, or 5, but for the Computer Science AP Exam, we require a 4 or 5 for credit for CSE 142.

F. Graduation Requirements

Summarize the graduation requirements for the program and the process for ensuring and documenting that each graduate completes all graduation requirements for the program. If applicable, describe the process for how course deviations are handled to ensure that graduation requirements are met. State the name of the degree awarded (Master of Science in Safety Sciences, Bachelor of Technology, Bachelor of Science in Computer Science, Bachelor of Science in Electrical Engineering, etc.)

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 systems 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 meet regularly with the advising staff. A special checkpoint takes place two to 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 process includes a plan of courses in progress and those scheduled for the remaining quarters leading to graduation. The result is that each student is required to have an acceptable terminal course plan over two quarters in advance of their actual graduation.

G. Transcripts of Recent Students

The program will provide transcripts from some of the most recent graduates to the visiting team along with any needed explanation of how the transcripts are to be interpreted. **These transcripts will be requested separately by the Team Chair.** State how the program and any program options are designated on the transcript. (See 2019-2020APPM, Section I.E.3.a.)

CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES

A. Mission Statement

Provide the institutional mission statement.

Mission Statement—UW

The primary mission of the University of Washington is the preservation, advancement, and dissemination of knowledge. The University preserves knowledge through its libraries and collections, its courses, and the scholarship of its faculty. It advances new knowledge through many forms of research, inquiry and discussion; and disseminates it through the classroom and the laboratory, scholarly exchanges, creative practice, international education, and public service. As one of the nation's outstanding teaching and research institutions, the University is committed to maintaining an environment for objectivity and imaginative inquiry and for the original scholarship and research that ensure the production of new knowledge in the free exchange of facts, theories, and ideas.

To promote their capacity to make humane and informed decisions, the University fosters an environment in which its students can develop mature and independent judgment and an appreciation of the range and diversity of human achievement. The University cultivates in its students both critical thinking and the effective articulation of that thinking.

As an integral part of a large and diverse community, the University seeks broad representation of and encourages sustained participation in that community by its students, its faculty, and its staff. It serves both non-traditional and traditional students. Through its three-campus system and through continuing education and distance learning, it extends educational opportunities to many who would not otherwise have access to them.

B. Program Educational Objectives

List the program educational objectives and state where they may be found by the general public as required by APPM Section I.A.6.a.

Our mission statement leads to the following program educational objectives, which are available online: 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

Describe how the program educational objectives are consistent with the mission of the institution.

The departmental program objectives are completely consistent with those of the University of Washington. 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. Our program objectives address both the education and impact aspects of that goal.

D. Program Constituencies

List the program constituencies. Describe how the program educational objectives meet the needs of these constituencies.

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

- Computer Engineering students
- CSE faculty

For students, our educational objectives enable them to use computer engineering as they lead impactful, meaningful lives and successful careers.

For faculty, our educational objectives are an essential motivation for pursuing academic careers. Our strong and successful students and alumni help recruit, retain, and motivate our world-class faculty.

E. Process for Revision of the Program Educational Objectives

Describe the process that periodically reviews the program educational objectives including how the program's various constituencies are involved in this process. Describe how this process is systematically utilized to ensure that the program's educational objectives remain consistent with the institutional mission, the program constituents' needs and these Criteria.

Our Program Educational Objectives have remained stable and have withstood the test of time, so reviews are typically brief. However, we do have several processes in place to ensure periodic review.

First, any substantial curriculum change is considered in light of our program educational objectives. As described later in this report, our substantial revision and alignment with Electrical Engineering courses in 2015 was performed in this context. At that time, the department's Executive Committee reviewed our program educational objectives and did not change them.

Second, any change to the institutional mission causes us to revisit our own mission as we are an integral part of the University of Washington. As discussed above, our objectives are fully aligned with UW's mission.

Third, in Fall 2017, our students initiated a conversation with our advising staff about creating a Student Advisory Council (<https://sac.cs.washington.edu/>). This advisory council has both elected and appointed members who represent a diverse set of interests among the undergraduate population. The council's main objective is to improve the overall experience of undergraduates in the Allen School. The council hosts presentations in addition to gathering feedback to pass on to the Allen School administrative units. This group has been a huge asset as we continue to work on improving our program. The inherent turn-over among student leadership on an annual basis provides a natural opportunity to review our program educational objectives with this constituency.

Finally, we seek validation for our objectives from outside our constituencies, primarily through employers of our graduates and our alumni. The fact that we are one of the top suppliers in the nation to Amazon, Microsoft, Google, and Facebook, as well as being the predominate supplier to small and mid-size companies in the Puget Sound region, means that program leadership is in close touch with recruiting managers and engineers at these companies, providing a tight feedback loop. A full-time external relations director provides a formal connection to over 100 companies in our Industrial Affiliates program.

CRITERION 3. STUDENT OUTCOMES

A. Student Outcomes

List the student outcomes and state where they may be found by the general public as required by APPM Section I.A.6.a. If the student outcomes used by the program are stated differently than those listed in Criterion 3, provide a mapping of the program's student outcomes to the student outcomes (1) through (7) listed in Criterion 3.

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:

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
3. an ability to communicate effectively with a range of audiences
4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

B. Relationship of Student Outcomes to Program Educational Objectives

Describe how the student outcomes prepare graduates to attain the 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 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:

2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
3. an ability to communicate effectively with a range of audiences
4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

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 while setting a lower bar on leadership activities. Most particularly relevant outcomes are:

2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

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:

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

CRITERION 4. CONTINUOUS IMPROVEMENT

A. *Student Outcomes*

It is recommended that this section include (a table may be used to present this information):

1. A listing and description of the assessment processes used to gather the data upon which the evaluation of each student outcome is based. Examples of data collection processes may include, but are not limited to, specific exam questions, student portfolios, internally developed assessment exams, senior project presentations, nationally-normed exams, oral exams, focus groups, industrial advisory committee meetings, or other processes that are relevant and appropriate to the program.
2. The frequency with which these assessment processes are carried out
3. The expected level of attainment for each of the student outcomes
4. Summaries of the results of the evaluation process and an analysis illustrating the extent to which each of the student outcomes is being attained
5. How the results are documented and maintained

A.1 *Data Sources*

A listing and description of the assessment processes used to gather the data upon which the evaluation of each student outcome is based. Examples of data collection processes may include, but are not limited to, specific exam questions, student portfolios, internally developed assessment exams, senior project presentations, nationally-normed exams, oral exams, focus groups, industrial advisory committee meetings, or other processes that are relevant and appropriate to the program.

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 both of our constituencies, as well as employers, alumni, and advising staff. 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

Assessment Process	Description	Frequency	Where Maintained
Regular Course Work	Instructors establish, assess, and evaluate performance indicators.	Regularly	Online on CSE servers
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
Exit Survey	Administered by CSE	Yearly	CSE Advising
Exit Survey	Administered by COE	Regularly	UW Office of Educational Assessment and CSE Director's Office

<i>Assessment Process</i>	<i>Description</i>	<i>Frequency</i>	<i>Where Maintained</i>
Targeted Course Work	Individual assignments and exams, or portions of assignments or exams, may be targeted to assess particular outcomes.	Regularly	UW Office of Educational Assessment and CSE Chair's Office
Capstone Courses	Capstones are showcases for many student outcomes.	Regularly	CSE Instructors; Dept. Videos
Per-Course Instructor Hand-Off Documents	Many of our courses, particularly at the 300-level, have multiple instructors each year, who regularly hand-off materials and "to do" lists	Yearly	Varies by course due to differences in infrastructure and tooling.
Full Faculty Approvals	All curriculum changes (requirements, pre-requisites) are approved by the full faculty, usually via unanimous consent after any helpful discussion	As needed	Emails to faculty mailing list
Curriculum Committee	Department level examination of multi-course portions of the curriculum. Approves all curricular changes (e.g., requirements, pre-requisites) before seeking full faculty approval	Weekly	Google Drive for the Committee
Special Case Reviews	In 2018-2019, a committee of senior faculty reviewed our 100-level course offerings and their relationship to admissions and the major, producing a substantial report with over 30 recommendations	Various	Final report issued in May 2019 to the CSE administration, curriculum committees, relevant faculty, and other campus stakeholders
Student Advisory Council	A student led council that plans presentations for students and gathers feedback from students.	Quarterly	SAC student organization and reports are maintained by the student org
Ugrad Lunch with the Director	Open meeting with the Director to hear about the School news and ask questions	Quarterly	Advising Notes in Team Drive
Climate Study	Survey of all CSE majors and pre majors	Annually	Advising and Admin Files
Anonymous Feedback	A portal for current students to give feedback anonymously to the advising staff	Various	Advising notes in Team Drive

Table 4-2 Assessment procedures for student outcomes

<i>Student Outcome</i>	<i>Primary Data Sources</i>	<i>Significant Additional Sources</i>
1. an ability to identify, formulate, and solve complex engineering problems by applying	Regular Course Work Targeted Course Work Capstones	Alumni survey Employer feedback

<i>Student Outcome</i>	<i>Primary Data Sources</i>	<i>Significant Additional Sources</i>
principles of engineering, science, and mathematics		Per-Course Instructor Hand-Off Documents Instructor Reflections
2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors	Targeted Course Work Capstones	Employer feedback Instructor Reflections
3. an ability to communicate effectively with a range of audiences	Targeted Course Work Capstones	Employer feedback Alumni survey
4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts	Targeted Course Work Capstones	Employer feedback Exit Surveys
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives	Targeted Course Work Capstones	Alumni survey Climate Studies
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions	Targeted Course Work Capstones	Alumni survey
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.	Targeted Course Work Capstones	Employer feedback Alumni survey

A.2 Assessment Process Overview

The frequency with which these assessment processes are carried out

Figure 4-1 gives a high level overview of our assessment and evaluation procedures. Here we provide a more global view of how these pieces interact with clear feedback loops that guide how we engage with the data and use it to drive continuous improvement.

The flow-chart below also includes two items not covered by the sources in Table 4-1:

1. *External Advisory Committee*: Every UW program must undergo a review every ten years in which an external committee reviews all aspects, including curricular goals and outcomes. Beyond that, we had maintained a standing advisory committee, but in the last few years have experimented with replacing that static body with a set of external focus groups (alumni, employers, senior technology leaders, etc.). A set of groups held meetings in 2014-2015 and notes were taken by School leaders. While useful at the time, we concluded the effort was burdensome and *ad hoc*, so we plan to create a new external advisory committee (returning to the old model) in the coming year.

2. Student course evaluations, managed by the standard UW processes, provide quantitative and qualitative information on each course. Per UW rules, quantitative results are shared with both instructors and School leadership, but qualitative results are shared only with instructors, who can (and do) use them to inform their self-reflections and instructor hand-offs, as described in Table 4-1.

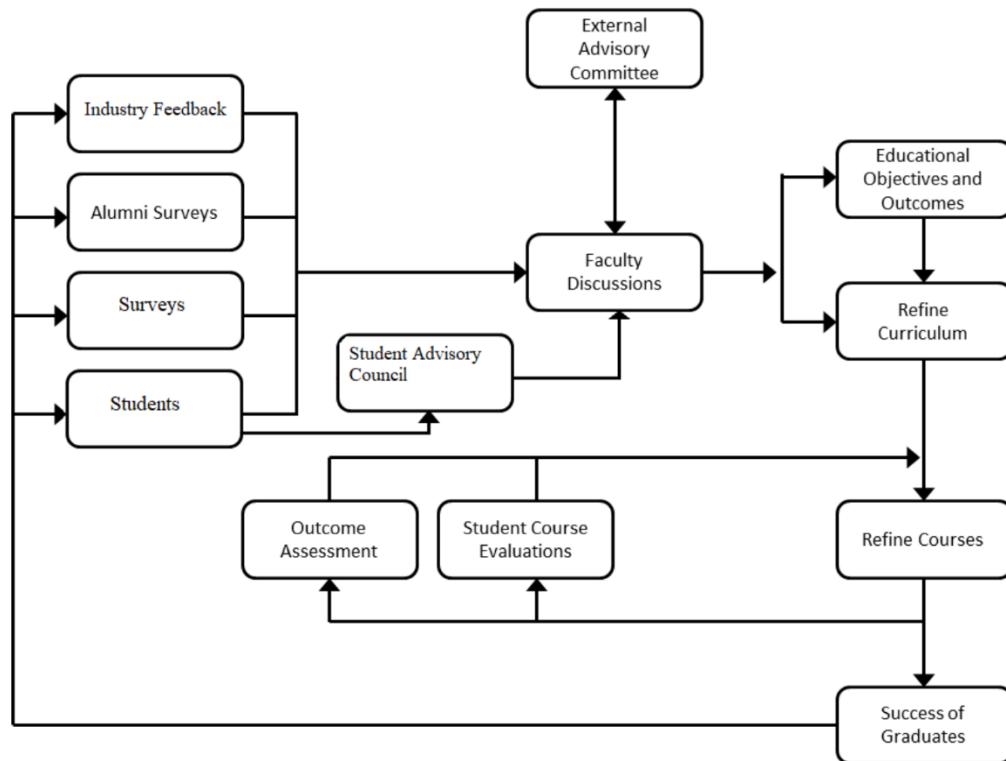


Figure 4-1 Summary of Department Assessment and Improvement Processes

There are two primary feedback loops in our process, one best suited to quickly refining individual courses and our program, 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. This loop uses feedback from current students, graduating students, alumni and our industrial affiliates. We used to ask an outside group, the Center for Instructional Development and Research (CIDR), to interview our graduating students. Their office has recently lost a number of employees, so at this point in time we are handling these evaluations internally. Additionally, we run an exit survey that includes both qualitative and quantitative assessments of our program by our graduating students, as well as alumni surveys soliciting feedback from our students after they have experience in the workplace or graduate school. Finally, we get feedback from our

Affiliates and co-op employers 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, the Undergraduate Advising Office, and the Curriculum Committee continually evaluate the overall program. There is no set timetable for evaluation – it is ongoing. The staff of the Advising Office meets weekly with the Undergraduate Program Coordinator to discuss issues that have arisen. The Curriculum Committee meets weekly 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. 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.

In addition, the recently constituted Student Advisory Council (SAC) has been a significant addition to our feedback loop process. This group of students helps gather concerns from our rapidly growing student body and helps us determine what issues need the most urgent attention. SAC is comprised by directors elected democratically by our student body and at-large members that represent the other Allen School student groups and certain student populations. SAC meets weekly during the academic year and hosts multiple discussion forums per quarter for students to build community and express their ideas and opinions. Most discussion forums include Allen School faculty and staff as panelists. Topics that SAC have recently covered include vulnerability, Allen School admission policies, and accessibility.

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. 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 three categories: High, Medium, and Low.

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. 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 Director 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 Director 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 resolution.

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. We also have extensive outreach activities at the K-12 level.

A.3 Targeted Assessment of Outcomes in Courses

The expected level of attainment for each of the student outcomes

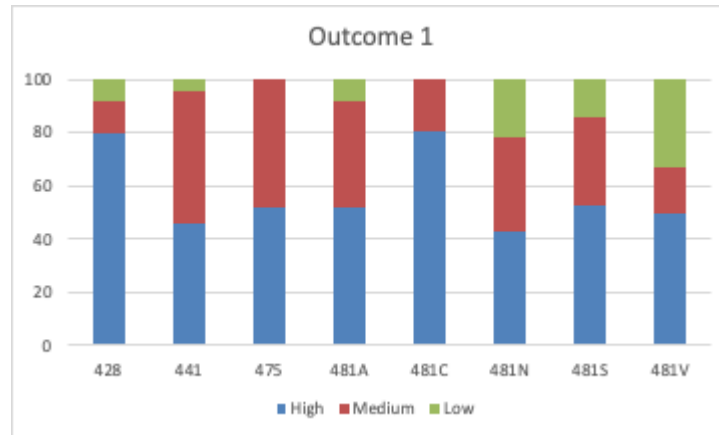
The most direct assessment we perform of student outcomes is through targeted assessment in courses. Instructors define performance indicators for the student outcomes addressed by their courses. They then use those indicators to categorize student achievement in three levels: High, Medium and Low. The 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. In evaluating the assessments, our standard is an average outcome of at least 80% in the High or Medium 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.

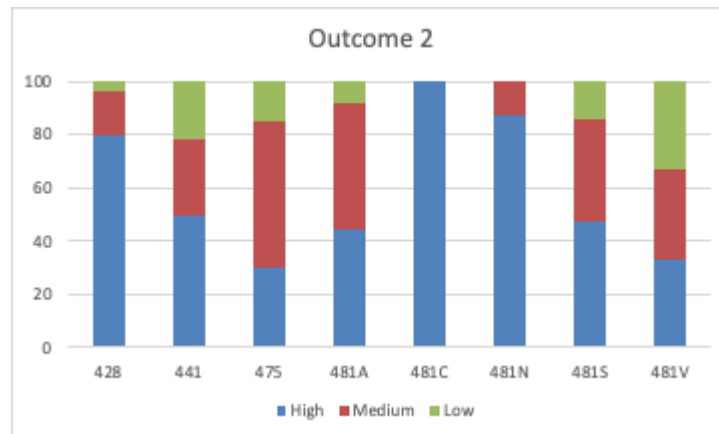
Over the past two years, we have used the capstone courses as the primary targets for evaluating student outcomes. The capstone courses serve a critical role in our program. In these courses, the students work in teams to accomplish a substantial project that covers multiple facets such as design, implementation, and evaluation. Each of our capstones has components that directly address the targeted outcomes. In performing the evaluation of a specific outcome, the instructors consider all the student submissions for an assignment or course work that is designed to address the outcome. At the end of the year, the results from these evaluations are distributed to all the capstone course instructors so as to inform them of the state of the entire program and help them refine the next iteration of the course offerings.

We present below the results gathered by the instructors for the most recent incarnation of the capstone courses offered in the last two years.

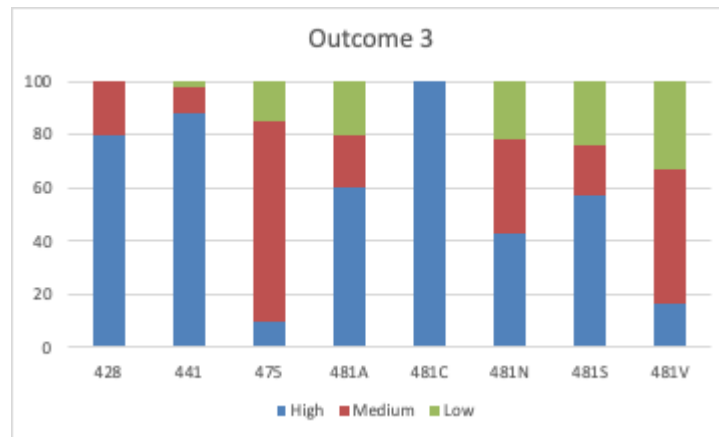
1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics



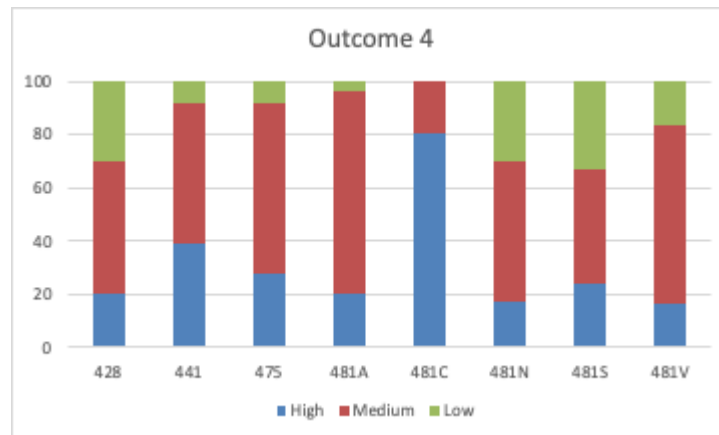
2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors



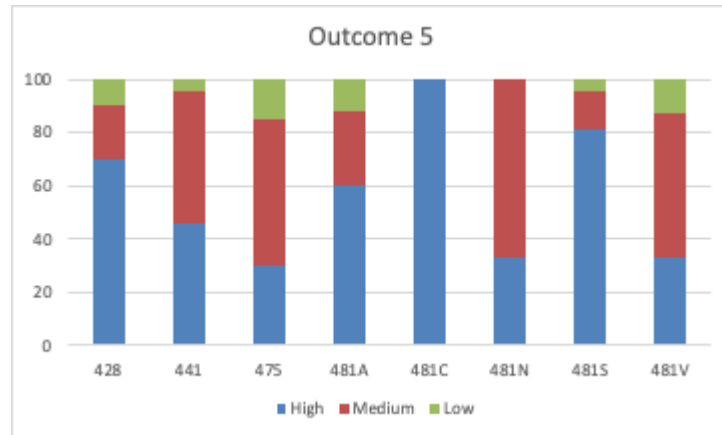
3. an ability to communicate effectively with a range of audiences



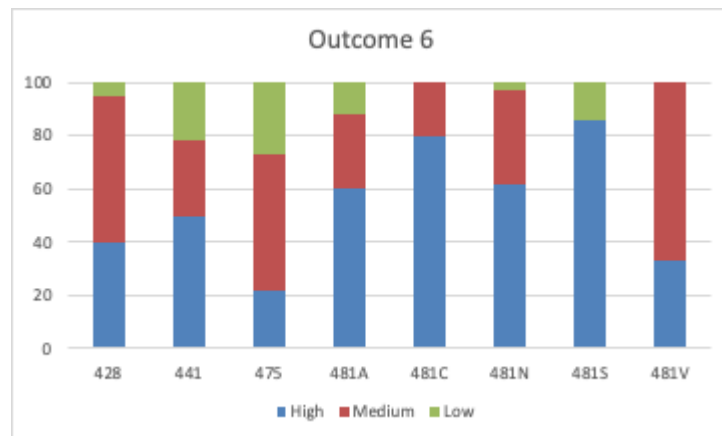
4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts



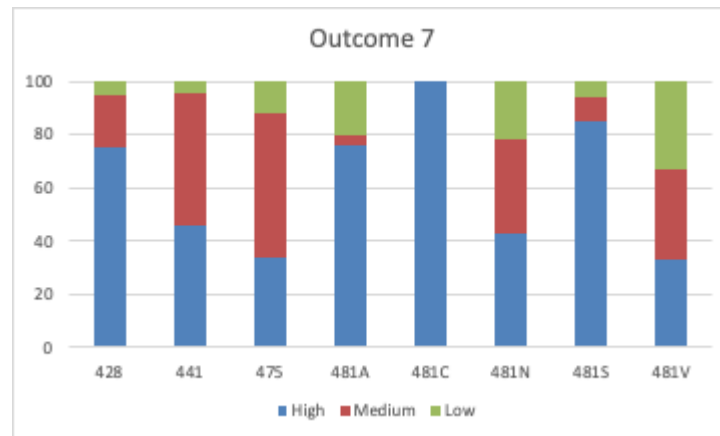
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives



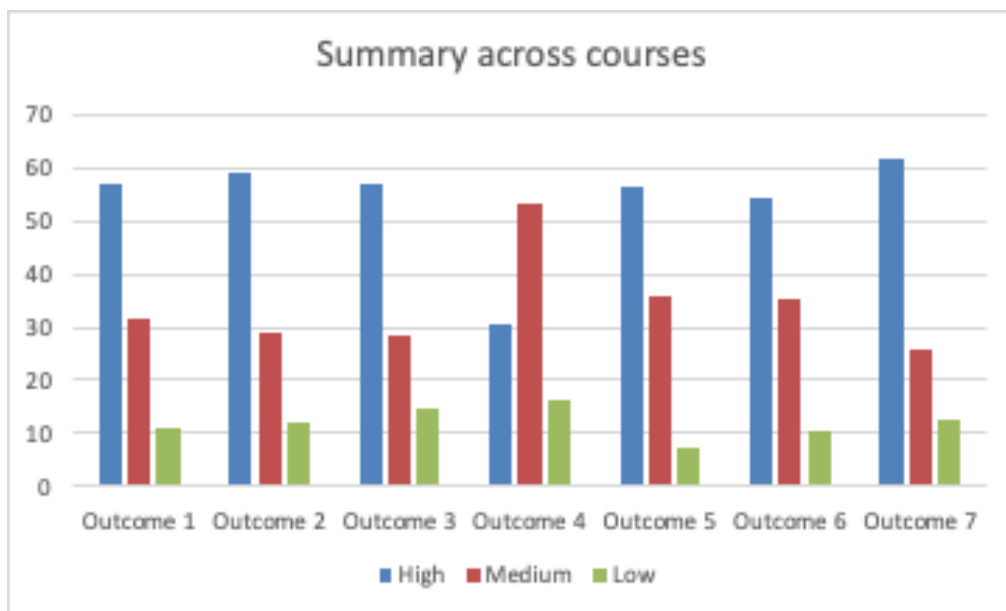
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions



7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.



The following chart summarizes the results of the different outcomes across the various capstones.



Looking over these results we see that for the most part we are meeting our goal that about 80% of our students achieve at the High or Medium level. In fact, the dominant category of student performance is the High category, which accounts for more than 50% of the students for most outcomes. One particular exception is Outcome 4, which takes into account the students' ability to recognize ethical and professional responsibilities in engineering situations. This finding is consistent with the end of program interviews with students (see A.4.3 below). While we have experimented with several seminars and in-course ethics examples in the past, the curriculum

committee is currently analyzing the data and proposing next steps on how to further incorporate a comprehensive ethics curriculum into our program. We are hoping it will be a mix of embedded modules into existing courses and seminars that dive deeper into this very important area of study.

A.4 Additional Assessment Processes

A.4.1 Employer and Alumni 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, Facebook, Amazon, and others - http://www.cs.washington.edu/industrial_affiliates/current/). These major and local companies hire the bulk of our students. We take advantage of our Industrial Affiliates Program to obtain feedback on our students and our career-readiness activities.

We also have regular alumni meet-ups at various companies as well as open events both in Seattle and in the Bay Area. We receive feedback on the performance of our students through these gatherings. These happen several times each 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 employees; and are you seeing things in students from other institutions that might suggest we should look at their curricula for potential improvements to ours?

A.4.2 Co-op feedback

In this section, we go over the survey results from the students and employers that participate in a co-op. Approximately 70% of our students participate in at least one internship and many of them choose to pursue co-op credit. Students and employers fill out a survey with the following questions. Students fill it out twice, evaluating their own skills before the internship and after the internship. Employers fill it out one time after their student completes the internship. Since these were before we moved to the new 1-7 ABET questions, this references the older a-k questions. See Table 4-5

Table 4-5 Co-op Survey Categories

- | |
|--|
| <ol style="list-style-type: none">1. Ability to apply knowledge of mathematics, science, and engineering2. Ability to design and conduct experiments, as well as to analyze and interpret data3. 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 sustainability4. Ability to function on multi-disciplinary teams5. Ability to identify, formulate, and solve computer engineering problems6. An understanding of professional and ethical responsibility7. Ability to communicate effectively8. Ability to understand the impact of computer engineering solutions in a global, economic, environmental, and societal context9. A recognition of the need for, and an ability to engage in life-long learning |
|--|

10. Knowledge of contemporary issues
11. Ability to use the techniques, skills, and modern computer engineering tools necessary for engineering practice
12. Ability to think critically
13. Ability to manage change
14. Understanding of office dynamics

Figure [4-1] shows average survey results for co-ops that took place between Autumn 2017 and Summer 2018. Figure [4-2] shows average survey results for co-ops that took place between Autumn 2018 and Winter 2019 (note that we did not have any CE students participate in a co-op during Spring of 2019 and Summer 2019 data is not available as of the submission of this report). Both of these graphs show similar trends. The first observation is that students do perceive an improvement in each of their surveyed categories upon the completion of their internship (varying from incremental to significant improvement). The second observation is except on two occasions, employers perceive students to be performing as well as, or better, than students perceive themselves to be performing upon completion of their internship.

2017-2018 Student & Employer Evaluation Survey Data

Fall 2017 - Summer 2018

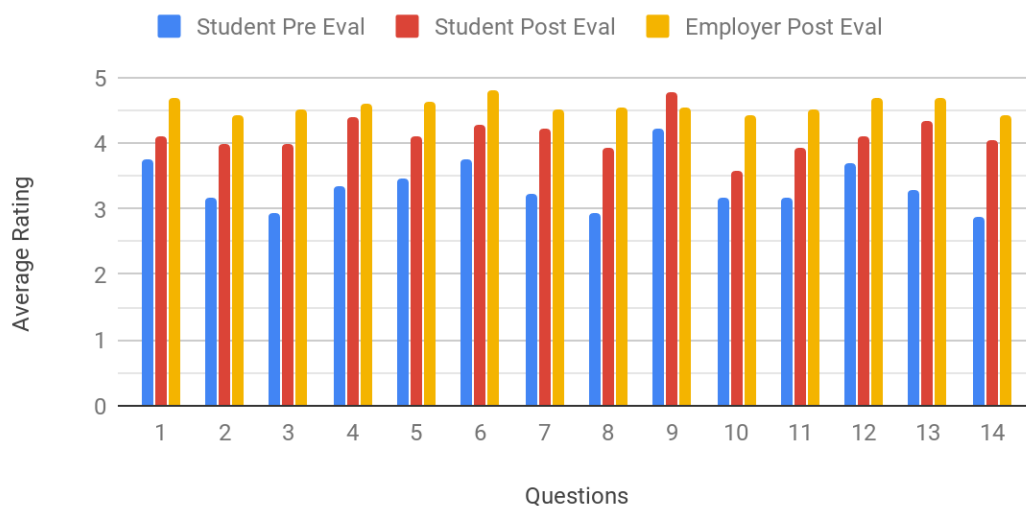


Figure 4-1 Co-Op Evaluation Data 2017-2018

2018-2019 Student & Employer Evaluation Survey Data

Fall 2018 - Winter 2019

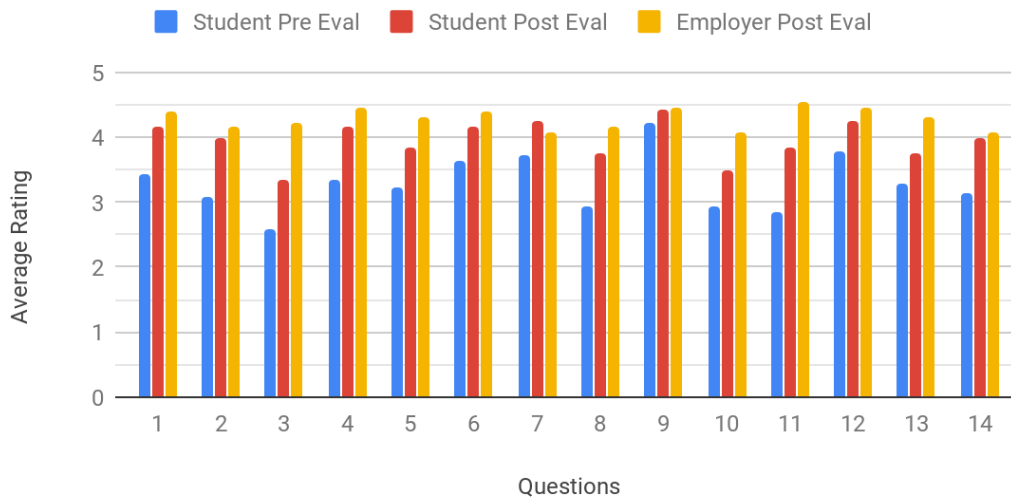


Figure 4-2 Co-Op Evaluation Data 2018-2019

In addition to the quantitative data just shown, we also make use of the free form comments made by employers. We cite here some typical examples the 2017-2019 surveys, selected because they reflect directly on our four program educational objectives, as listed in Section 2.B.

Engineering Quality

They surprise me by their ability to judge things, identify compromises and make decision. They are definitely not a typical intern and demonstrate an unexpected level of maturity for somebody relatively new at computer sciences

On quality of work: Notable for a solid transfer of knowledge at the end of internship to remaining team.

They were able to work on the full project phases from design to implementation to testing.

Even though they were unfamiliar with either development environment, they wrote code that followed best practices in those areas. They sometimes reworked code after feedback.

Leadership

I have witnessed them helping others on numerous occasions. Everybody likes working with them.

They worked independently and sought help from subject matter experts when it was needed. They, along with the other interns, had great working relationships. They, along with the other interns, designed a spirit contest aiming to boost camaraderie.

They always do what they say they are going to do, despite the fact that our environment is pretty challenging with high priority disrupting tasks almost every day.

Economic Impact

They completed projects for us that will help us do integration testing, and a service that allows us to do feature toggles and A/B testing.

They worked well with the team. Required minimal supervision. Knew when to ask questions, which is very important. Produced tools that the team will use for a long time after he is gone. Very satisfying outcome all around.

Lifelong Learning

Not only are they always interested in expanding their scope and learning new things, they are also particularly good at switching with minimal ramp up time.

They were a very independent learner and did not need any remedial instruction.

A.4.3 End-of-program interviews

From 2005 through 2015, we asked staff from the UW Center for Instructional Development and Research (CIDR) to interview our soon-to-be-graduating students. Current students were 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 felt that the online surveys gave a more accurate picture of student sentiment since it eliminated social pressure that might arise to agree with the majority in a group meeting.

The questions CIDR asked (and that we now ask in our alumni survey) 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 (1-7) (Section 3.A). Students rank the contribution on a scale of 0 to 5. The student responses for the most recent two years are shown below in Figure 4-3.

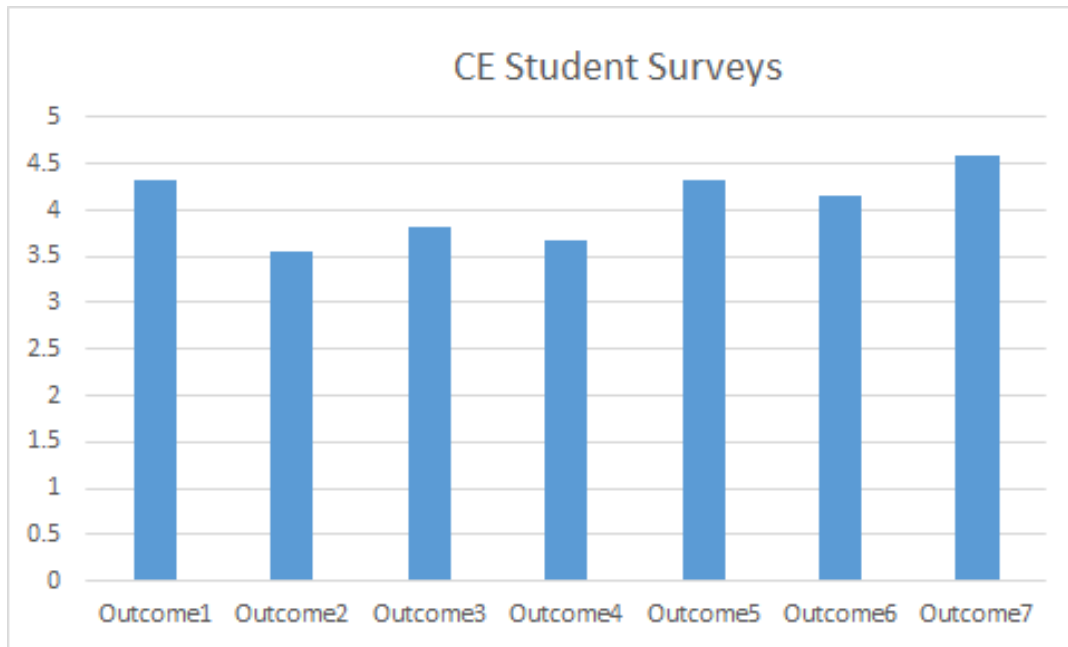


Figure 4-3 Student assessments of outcomes from end-of-program surveys

Our weakest areas are:

- Outcomes 2: an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
- Outcome 3: an ability to communicate effectively with a range of audiences
- Outcome 4: an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts

A.4.4 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.

Plans After Graduation (Class of 2019)

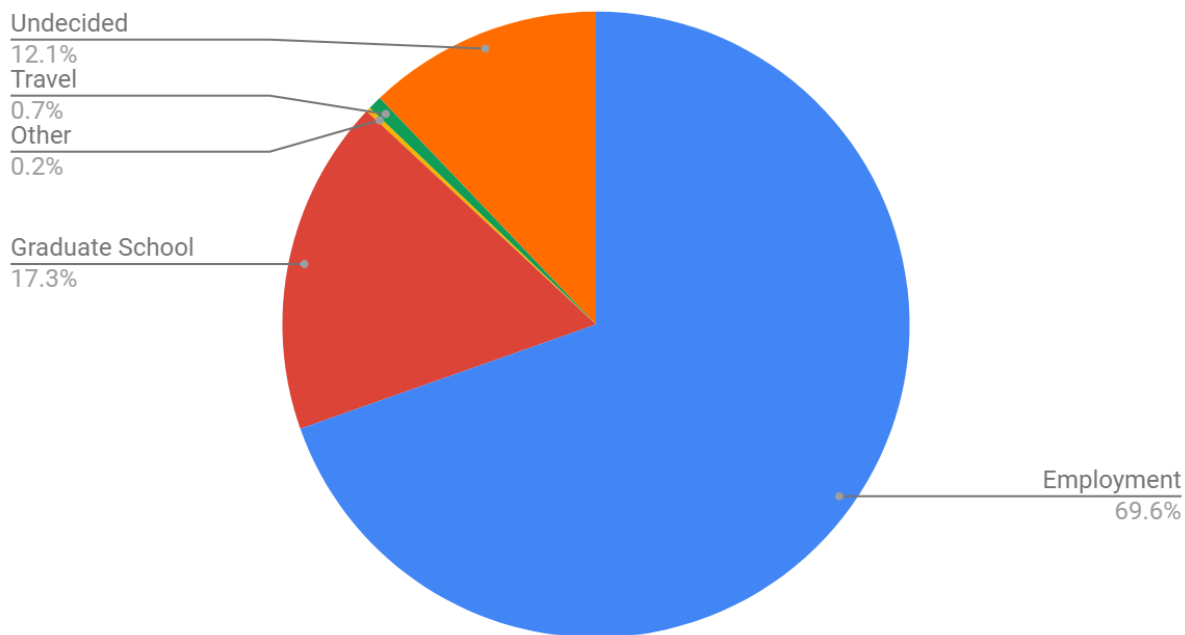


Figure 4-4 Graduate Exit Survey Company Destination Data for Undergraduates (2019)

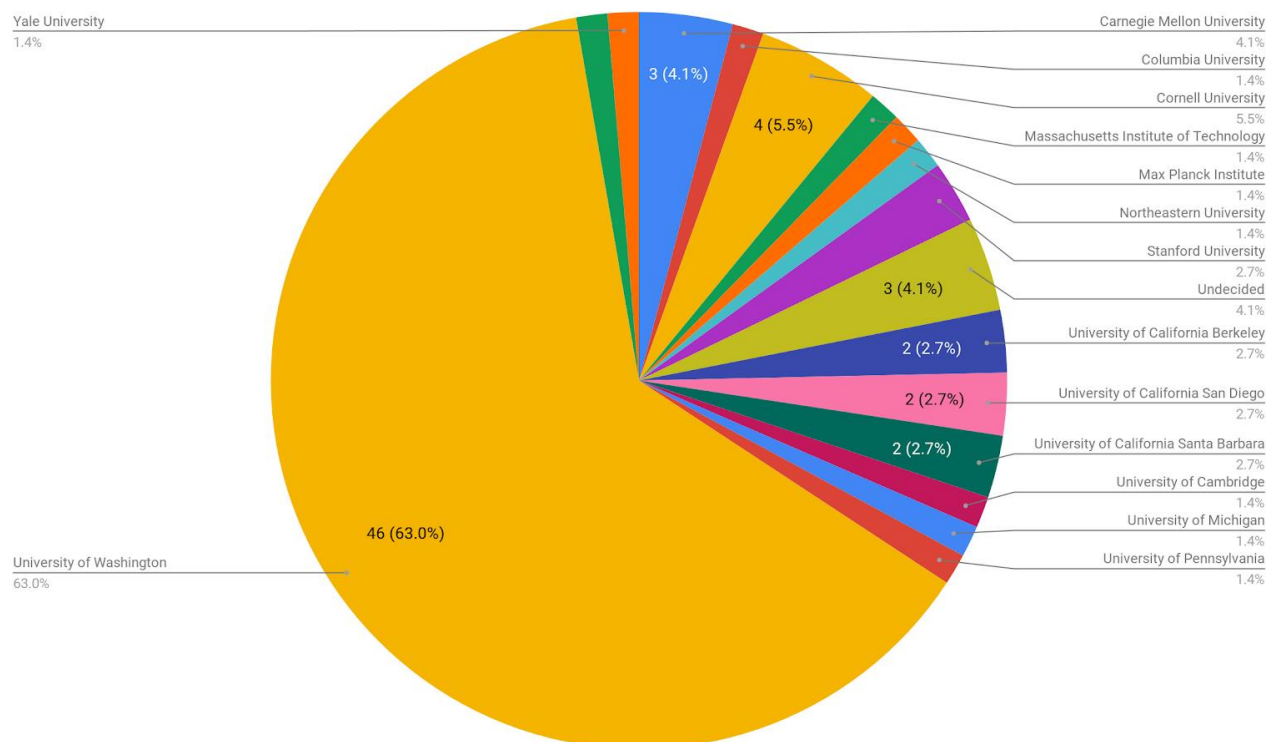


Figure 4-5 Graduate Exit Survey Destination Data for Undergraduates (2019)

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 Director of Student Services and reported to the Undergraduate Program Coordinator and the Allen School Director. 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 the quality of their education, the faculty and advising. Some students expressed that the wait on email responses was a bit longer than hoped for during peak times. Students also asked for more courses on cloud computing, parallel computing and computational biology.

A.4.5 Student course evaluations

During course evaluations, students 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.

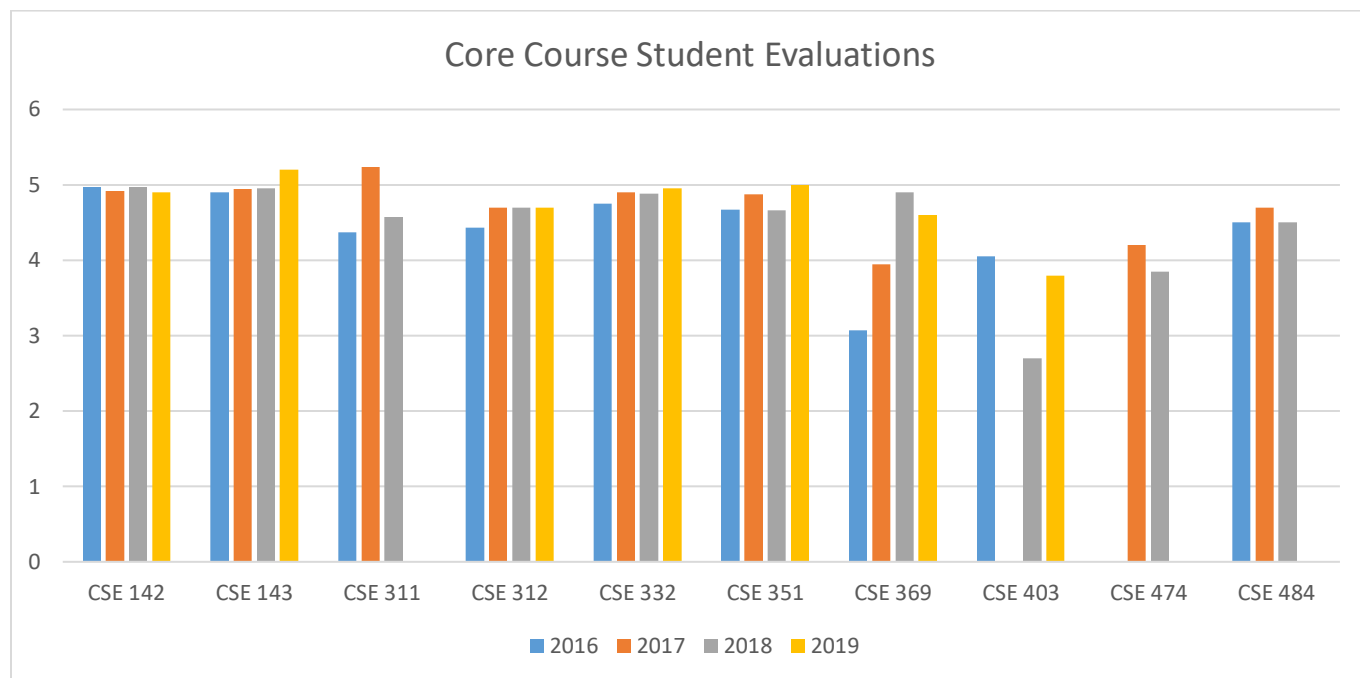


Figure 4-6 Core Course Student Evaluations (2016-2019)

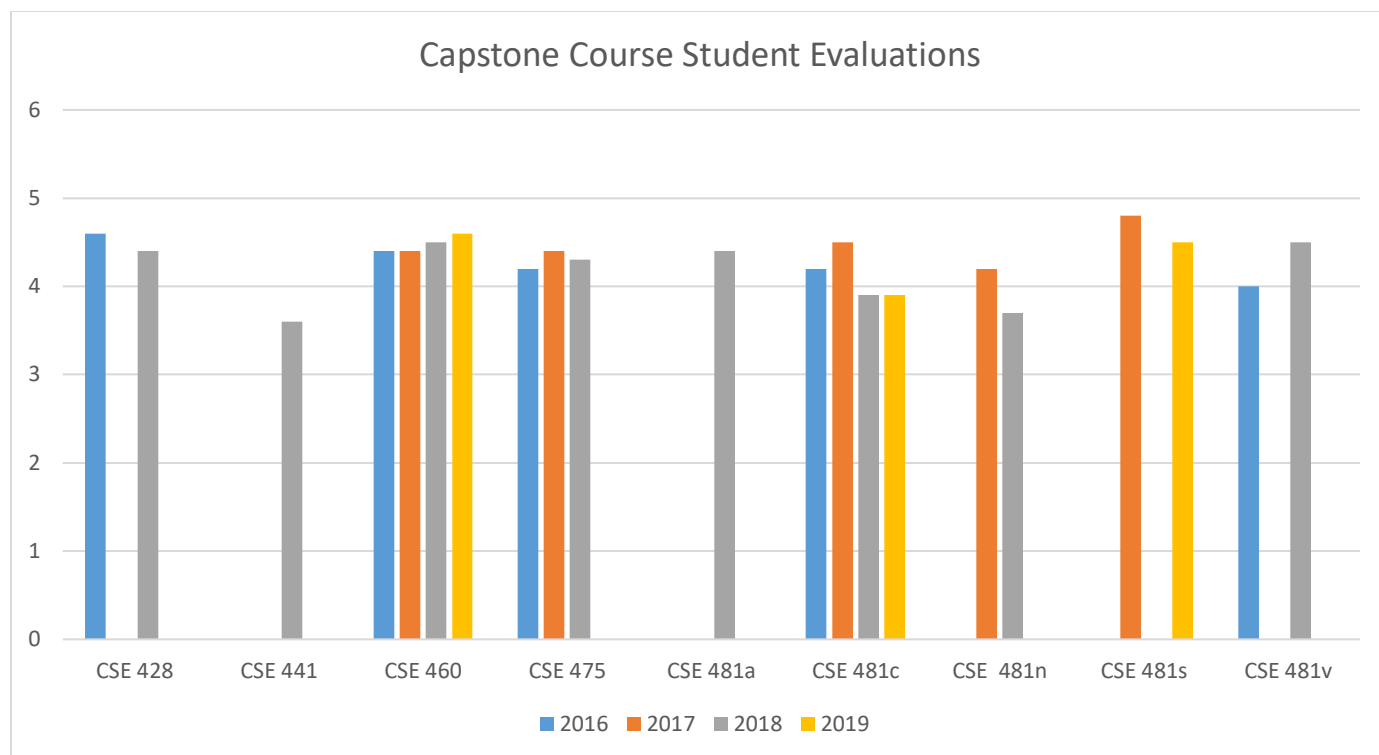


Figure 4-7 Capstone Course Student Evaluations (2016-2019)

Student rankings are on a five point scale, but because the university applies a normalization function that takes into account the size, difficulty, and workload of a course, “adjusted” scores can be somewhat higher than five. Figures 4-6 and 4-7 show the adjusted student ratings for each offering of each core course and each capstone course. The x-axis has the average adjusted mean score for all offerings of a course in a particular year, and the y-axis is the adjusted mean overall score for that offering.

The results are generally quite good. Courses that warrant monitoring are typically ones taught infrequently and by only one or two faculty members, a combination that can sometimes make updating a course more of a struggle than it would be in the typical case.

A.4.6 Student Anonymous Feedback

In 2015, we built an anonymous-feedback tool to allow students in any of our courses to provide feedback to their instructor anonymously. In 2018, we extended this tool in two directions: to allow students to also direct their feedback to Allen School academic leadership and/or the Director of Student Services, and to allow TAs to use the tool for the course they are TAing. Despite advertising these new feedback channels, so far these extensions have seen only modest use – a handful of messages with a couple complaining about grading or instructor interactions and a couple raving about excellent faculty. We need to continue publicizing this feedback channel, but its modest use is probably a good sign in and of itself.

A.4.7 TA Review

In 2018, we also rolled out two new review mechanisms related to Teaching Assistants, an evaluation of TAs by instructors and a lightweight survey for TAs to provide feedback about our courses. Both mechanisms have results reviewed by Allen School leadership and fed back to instructors to guide continuous improvement. Both review mechanisms have had extremely positive results: Instructors indicate our TAs are excellent and TAs indicate the courses and their TAing experience are excellent.

A.4.8 Faculty self-evaluations

Our full-time faculty, both tenure-track and teaching (lecturers), go through a merit-review process annually. For junior faculty, there is a much more rigorous annual review process in which faculty members prepare self-assessment materials, a faculty committee reviews them and generates a report, and faculty of more-senior ranks discuss and identify feedback delivered via the Director. Course instruction is a key part of this review, considering not only student evaluations, but curricular innovations, the faculty member's self-reflection, and peer classroom observations.

Many of our faculty also regularly utilize the College of Engineering's Center for Teaching and Learning, which has a well-honed procedure for visiting classrooms during terms to receive feedback from students, producing quantitative and qualitative reports that are shared back with the faculty member.

A.4.9 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, to the department as a whole, and to the public.

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 1 through 7. It is common for final project demonstrations to be open to the public. 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. Some of these videos are available online at <https://www.youtube.com/user/UWCSE/videos>.

A.4.10 Curriculum Committee

An undergraduate curriculum committee, with several faculty as well as two members of the student services staff, regularly reviews all aspects of our curriculum – approving new courses, identifying curricular gaps, identifying ways to improve rough spots in student experience, etc. After a few years of trying to work electronically, in 2018 we moved to weekly meetings to increase significantly the quality and quantity of discussion and results. This has also led to clear and concise rationales for changes that are then sent to the full faculty for further discussion or consent. At each meeting, a written agenda is edited to form notes from the meeting and to evolve into the next week's agenda.

B. Continuous Improvement

Describe how the results of evaluation processes for the student outcomes and any other available information have been systematically used as input in the continuous improvement of the program. Describe the results of any changes (whether or not effective) in those cases where re-assessment of the results has been completed. Indicate any significant future program improvement plans based upon recent evaluations. Provide a brief rationale for each of these planned changes.

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

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

We have not made significant changes to this area and at this time appear to be hitting our targets in most courses. There are a few problem courses that have come to light recently and we plan to investigate these more thoroughly to see if we can pinpoint trouble areas and make improvements going forward.

2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors

*This has been a particular area of interest for students in our focus groups through the Student Advisory Committee in the last year. Recent surveys suggest that we could go further in this area. We have introduced new capstones in the last two years that directly tackle this issue, including our new CSE 482: **Capstone Software Design to Empower Underserved Populations**. Students work in teams to design and implement a software project involving multiple areas of the CSE curriculum, for the purpose of empowering marginalized or underserved populations. We now need to work harder to embed these ideas throughout the courses.*

3. an ability to communicate effectively with a range of audiences

Students always rate themselves low in this area, and could always use more practice. We have introduced a few different pathways to give students more experience in this area and the curriculum committee will be looking at suggesting more solid guidelines for capstone courses to follow in the future for best practice. One area to note is that when students go on

internship, their managers rate them quite a bit higher than they rate themselves, so they may be under-appreciating their abilities at least to some degree.

4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts

We have tried a number of approaches to integrating ethics into our curriculum, and continue to do so. We altered the curriculum to require students to take one of three courses that embed ethics curriculum to ensure all students are exposed to this important area. Our most recent assessments from alumni 2 to 3 years out show this is still a weak area. We have introduced a myriad of popular seminars on this subject and are narrowing in on a course of action this fall. We submitted a proposal in January to attempt to secure funding to launch a new, comprehensive ethics approach in our curriculum but that proposal was unfortunately not funded. The curriculum committee has had extensive discussion on this subject and hopes to have a new course to introduce this next year.

5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

In light of past assessments, we introduced project practicum for a pilot course, and we are still analyzing data to determine if we should move all capstones to a two quarter long experience. Initial information shows that it was indeed a better experience for the students to hit the ground running in the 2nd course if they had a 2 credit prep course prior to starting the larger projects.

6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions

This has been an important part of all of our capstones and continues to be so at this time. Recent assessments show that this is a strong area for our students. We have introduced a several machine learning courses in the last few years as this is a subject area that students are increasingly interested in and industry demand is sky-rocketing.

7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

Some of our 300 level courses now give students a bit more experience in this area. We continue to see high assessments for this outcome. Our alumni surveys also show that our students continue to pursue this knowledge after they graduate as can be seen by the large number of students who proceed on to graduate school.

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 Allen School Director. 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 Courses Introduced Since 2013

Course	Title
<i>Pre-majors Courses</i>	
CSE 160	Data Programming
CSE 163	Intermediate Data Programming
CSE 180	Introduction to Data Science
<i>Non-core Majors Courses</i>	
CSE 190a	Direct Admission Seminar
CSE 340	Interaction Programming
CSE 390t	Transfer Seminar
CSE 369*	Introduction to Digital Design
CSE 371*	Design of Digital Circuits and Systems
CSE 402	Design and Implementation of Domain-Specific Languages
CSE 442	Introduction to Data Visualization
CSE 447	Natural Language Processing
CSE 469*	Computer Architecture I
CSE 470*	Computer Architecture II
CSE 478	Autonomous Robots
CSE 490	Deep Learning (permanent number pending)
CSE 490	Cryptography (first offering Fall 2019, permanent number pending)
CSE 490	Tech for the developing world (offered Fall 2018, permanent number pending)
<i>Capstone Design Courses</i>	
CSE 481a	Operating systems Capstone
CSE 481g	Distributed Systems Capstone
CSE 481s	Security Capstone
CSE 481c	Robotics Capstone
CSE 481v	Virtual Reality Capstone
CSE 481n	Natural Language Processing Capstone
CSE 482	Capstone Software Design to Empower Underserved Populations

* Courses marked with (*) are new courses with new content, but also include content that overlap with retired courses (CSE 352, CSE 471). Not included above are course re-numberings to better align numbers with cross-listed EE courses as re-numberings do not reflect true innovation/improvement.

Greater Alignment with Electrical Engineering

In 2015, we completed a significant reorganization of core pieces of our computer engineering curriculum to align with Electrical Engineering. The result is:

1. More cross-listed courses, allowing students additional flexibility in offerings.

2. Better shared teaching resources across the departments, again allowing more offerings but also providing better alignment for the several joint faculty between the programs
3. Better use of facilities: sharing lab space for shared courses
4. More senior-level opportunities for students: Via shared cross-listed courses, more Computer Engineering students have the pre-requisites for senior-level EE courses and vice versa.
5. More rational division of material into courses and less redundancy across courses.

Full documentation of the rationale and changes from 2015 is available. Here we sketch the key pieces:

- We removed CSE 352, a hardware design course that was focused mostly on very basic processor design and which was a poor fit for students with no prior hardware-lab experience.
- We created CSE 369, a “bridge course” that leverages the Boolean logic material in CSE 311 to quickly get students hardware-lab experience. Basically, this 2-credit course is enough for students to gain the experience necessary for CSE/EE 371 rather than completing the 5-credit EE 271. CSE 369 does the EE 271 labs, providing shared knowledge.
- CSE/EE 371 cross-lists what was previously a CSE-only course. This is the most advanced required digital design course for Computer Engineering.
- CSE/EE 469 and CSE/EE 470 provide a two-course sequence in processor design. Prior to the cross-listing, we did not have the teaching resources to provide a two-course sequence.
- As mentioned above, this alignment enables Computer Engineers to have the pre-requisites for additional EE senior electives, notably courses in VLSI.

The process was a broad collaboration of many faculty in both the Allen School and Electrical Engineering with a careful cataloguing of material moved, added, or removed. Full “master syllabi” for all new or renumbered courses were compiled and submitted to the faculty for approval as well as then sent through the University’s processes for new courses and degree-requirement changes. Allen School faculty meetings reviewed the overall curriculum change twice before it was approved.

Other Program Changes

While adding over a dozen new courses and completely reorganizing our digital logic and computer architecture courses comprise the most substantial improvements since our previous review, other program changes are worth noting as well. We catalog them here.

1. *Ethics*: As described in Criterion 5, our educational outcomes related to ethics and professional responsibility are ensured through material in CSE 403, CSE 474, and CSE 484, but our students and faculty want both opportunities for deeper discussions of ethical

challenges our field is facing and additional opportunities to connect technical material to societal challenges. We have made progress on both fronts and hope to further cement this progress in the coming year or two. For deeper discussions, we have offered four seminars on ethics in the last two years. Two were fairly conventional discussion seminars where students prepared by reading about a topic (e.g., biased artificial intelligence or online privacy) and then discussed. The other two took a much broader view of the future of artificial intelligence, with readings and discussion ranging from philosophy to neuroscience and more. For additional connections, discussions of bias, fairness, and accessibility are now common in new courses, notably natural language processing and interaction programming. Plans for the next year include better cataloguing where these modules appear and encouraging them to happen more regularly.

2. *Project practicum*: We encourage students to learn through team-based projects, but we noticed and fixed a gap in our courses. We had course numbers for capstones (described in detail elsewhere) and for independent research, but not for team projects that may not be aimed toward novel research results. We fixed this by adding the CSE 495 course number. To date, we have used this number for two purposes: For faculty-advised projects that contribute meaningfully to open-source projects, and for a number of projects in which students have contributed to making technology more accessible for people with disabilities.
3. *CSE 332*: We made substantial changes to CSE 332, including all new projects and more focus on using modern software tools and processes.
4. *CSE 312, 332, 333 content shift*: At the time of our 2013 review, our 300-level courses were only 2-3 years old and “what fit where” was still solidifying as our first students taking these courses were graduating. In 2014 or so, we moved around some topics to improve 300-level course outcomes and balance student workload. Specifically:
 - Material on intractability and P vs. NP moved from 312 to 332
 - Material on concurrency moved from 332 to 333 (material on parallelism remained in 332)
5. *More seminars*: Our students benefit immensely from low-credit seminar courses that complement their technical, graded courses. In addition to the ethics seminars discussed above and existing seminars on entrepreneurship and on career-advice-from-alumni/ae, we have been adding new seminars more often. In the last two years, we have had seminars on software reliability engineering, career patterns for success, and the history of computing.
6. *End-of-year optional poster fair*: Many of our capstone courses as well as other project courses have open poster sessions at the end of the term that attract members of our community and give our students a chance to communicate their ideas. In 2017, we added an annual end-of-year fair open to all students from all courses in which we have a dozen or so alumni or other supporters come to interact with students and award a prize.
7. *TA training*: While we have long had a TA training for TAs in our large introductory courses (CSE 142 and CSE 143), in 2016 we created a complementary weekly training for undergraduate TAs in our other courses. The content of the training has been evolved and assessed over the last three years.

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

When	Concern	Source	Response
2014-2015	Students, faculty, and industry requested more data science options	Surveys	We added a data science track to our undergraduate CS program in 2016-2017 and courses that CE students take as electives
2014-2015	Student interest in ML, NLP and Visualization skyrocketed	Surveys and Industry Feedback	Created four new 400 level courses, Natural Language Processing, Machine Learning, Robotics, and Data Visualization
2014-2015	Students wanted a Diversity and Accessibility course	surveys	Introduced Capstone course that also met Diversity Requirement and Accessibility
2014-2015	Huge demand from non-majors	Surveys	We added CSE 160, 163, and 416 as permanent courses.
2014-2015	Students wanted more time to focus on capstone-like experiences	Surveys	We added CSE 495 a project practicum course so we can offer more non capstone project courses
2014-2015	Students wanted more hardware courses	Student feedback and faculty analysis	Autumn 2015 we solidified a change to our Computer Engineering curriculum to cross list several courses with the Electrical and Computer Engineering Department. We re-designed and re-numbered a series of courses that are now jointly listed as CSE/EE: 469, 470, 474, and 475.
2014-2015	Faculty expressed frustration at current course flow for CE intro courses	Faculty feedback	Removed CSE 352 and added two new courses, CSE 369 and 371, to give students a better pathway into the hardware side of Computer Engineering
2014-2015	Students were often confused about exact requirements/ ABET had concerns as well	Surveys, Student feedback	We instituted a re-design of the advanced CE requirements to include a set of System Electives.
2014-2015	Student and Admissions feedback, decisions could now be made sooner	Surveys, Feedback	We officially moved our Direct Admission and Transfer Admission selection process from the department to the UW Admissions Office. This stream-lined admissions and made it more consistent with UW (this changed again in 2018)
2016-2017	Students express wanting more social experiences	Surveys and feedback in person	Student Advisory council was created and Q++ student group was created
2018	Student and Admissions Feedback	Surveys, Student feedback	Moved Transfer Admissions back to the Allen School

When	Concern	Source	Response
2019	Students want more clarity for seminars and special topics	Student feedback during advising appts	Created CSE 492 course number for seminars so there is now a clear distinction between 490 special topics courses and 492 seminars that are credit/no credit
2019	Students want more clarity for seminars and special topics	Student feedback during advising appts	Created CSE 492 course number for seminars so there is now a clear distinction between 490 special topics courses and 492 seminars that are credit/no credit

C. Additional Information

Copies of any of the assessment instruments or materials referenced in 4.A and 4.B must be available for review at the time of the visit. Other information such as minutes from meetings where the assessment results were evaluated and where recommendations for action were made could also be included.

These documents will be available for review during the visit.

CRITERION 5. CURRICULUM

Since our last ABET program review in 2013, the Allen School has designed and implemented an update to our core curriculum. We changed how the senior electives were distributed in order to ensure all students graduating with a Computer Engineering degree met the necessary engineering course requirements. We also introduced courses co-listed with Electrical and Computer Engineering (ECE) to broaden our course offerings in hardware 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 2015. During a transition period we offered both the old and new courses. We completed introducing the new set of courses during the 2016-17 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.

Effect of the Revision

We have had two sets of graduation requirements for our computer engineering degree since Autumn of 2013. 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:

Spring 2013 – Spring 2015	https://s3-us-west-2.amazonaws.com/www-cse-public/ugrad/curriculum/CompE_wi14.pdf
Autumn 2015 - present	https://s3-us-west-2.amazonaws.com/www-cse-public/ugrad/curriculum/CompE_au16.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 Computer Engineering component we distinguish between Required courses, CSE System Elective courses, and what we will call CSE General Electives. The former are required; there is some flexibility in the latter two (along the lines of “4 courses chosen from...”) while ensuring that every graduate has taken a set of courses that together meet our student outcomes.

A. Program Curriculum

A.1 Tables

Table 5-1 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 2019 & Sp 2019	657,248
	Natural Science Elective	S.E.	5	()			Wi 2019 & Sp 2019	175, 175
	English Composition	S.E.		()	5		Wi 2019 & Sp 2019	30,30
1 Yr/ 2 nd Qtr	Math 125 Calc w/Anal Geometry	R.	5	()			Wi 2019 & Sp 2019	1006, 442
	Phys 121	R.	5	()			Wi 2019 & Sp 2019	551, 546
	VLPA/I&S Elective	S.E.		()	5		Wi 2019 & Sp 2019	NA
1 Yr/ 3 rd Qtr	Math 126 Calc w/Anal Geometry	R.	5	()			Wi 2019 & Sp 2019	1004, 827
	Physics 122	R.	5	()			Wi 2019 & Sp 2019	556,425
	VLPA/I&S Elective	S.E.		()	5		Wi 2019 & Sp 2019	NA
2 Yr/ 1 st Qtr	Natural Science Elective	S.E.	5	()			Wi 2019 & Sp 2019	175, 175

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 2019 & Sp 2019	894, 487
	VLPA/I&S Elective	S.E.		()	5		Wi 2019 & Sp 2019	NA
2 Yr/ 2 nd Qtr	Math 308 Linear Algebra	R.	3	()			Wi 2019 & Sp 2019	1011,644
	CSE 143 Computer Programming II	R.		5 (✓)			Wi 2019 & Sp 2019	823, 591
	VLPA/I&S Elective	S.E.		()	5		Wi 2019 & Sp 2019	NA
	Math Science from Approved List	S.E.	3	()			Wi 2019 & Sp 2019	50, 50
2 Yr/ 3 rd Qtr	EE 205 Introduction to Signal Conditioning/ EE 215 option for either one	R.		4 (✓)			Wi 2018 & Au 2018	23, 10
	ENGR 231 Intro. to Technical Writing	R.		()	3		Wi 2019 & Sp 2019	221, 223
	VLPA/I&S Elective	S.E.		()	5		Wi 2019 & Sp 2019	NA
	CSE 311 Foundations of Computing II	R.	2	2 ()			Au 2018 & Sp 2019	306,249
3 Yr/ 1 st Qtr	CSE 351 The Hardware/Software Interface	R.		4 (✓)			Wi 2019 & Sp 2019	155, 217

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 332 Data Structures	R.		4 (✓)			Au 2018 & Wi 2019	149,296
	VLPA/I&S Elective	S.E.		()	5		Wi 2019 & Sp 2019	NA
3 Yr/ 2 nd Qtr	CSE 312 Foundations Computing II	R.	2	2 ()			Wi 2019 & Sp 2019	266, 96
	UW Writing Course	R. (S.E.)		()	4		Wi 2019 & Sp 2019	NA
	Free Electives	E.				2	Wi 2019 & Sp 2019	NA
	CSE 369	R.		2 (✓)			Wi 2019 & Sp 2019	37, 33
3 Yr/ 3 rd Qtr	CSE Core Course	S.E.		4 (✓)			Wi 2019 & Sp 2019	130,200
	CSE Core Course	S.E.		4 (✓)			Wi 2019 & Sp 2019	130,200
	Free Elective	E.		()		7	Wi 2019 & Sp 2019	NA
	CSE 371	R		5			Wi 2019 & Sp 2019	8,16
4 Yr/ 1 st Qtr	One of CSE403, CSE 474 or CSE 484	S.E.		4 (✓)			Wi 2019 & Sp 2019	30, 146
	One of CE System Electives	S.E.		4 (✓)			Wi 2019 & Sp 2019	136,146

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		
	Free Elective	E.		()		6	Wi 2019 & Sp 2019	NA
4 Yr/ 2 nd Qtr	One of CE System Electives	S.E.		4 (✓)			Wi 2019 & Sp 2019	136,146
	One CE System Electives	S.E.		4 (✓)			Wi 2019 & Sp 2019	136,146
	Free Elective	E.		()		4	Wi 2019 & Sp 2019	NA
	Free Elective	E.		()		4	Wi 2019 & Sp 2019	NA
4 Yr/ 3 rd Qtr	Capstone Course	S.E.		5 (✓)			Wi 2019 & Sp 2019	24, 24
	CSE Senior Elective	S.E.		4 (✓)			Wi 2019 & Sp 2019	130,200
	CSE Senior Elective	S.E.		5 (✓)			Wi 2019 & Sp 2019	130,200
				()				
			45	70	42	23		
TOTALS-ABET BASIC-LEVEL REQUIREMENTS			45	67.5				

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		
OVERALL TOTAL CREDIT HOURS FOR COMPLETION OF THE PROGRAM			45	70	42	23		
PERCENT OF TOTAL			25%	39%				
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

Describe how the curriculum aligns with the 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)
- 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 our students must take CSE 351, the Hardware/Software Interface, as an introduction to this level of modern computer systems.

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 369 and CSE/EE 371. These courses involve a substantial laboratory component and hands-on design.

All students must take at least one of CSE 403 (Software Engineering), CSE 474 (Software for Embedded Systems), or CSE 484 (Security). These courses involve teamwork, a consideration of ethical responsibility, and an awareness of contemporary issues.

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 teamwork, 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

Describe how the curriculum and its associated prerequisite structure support the attainment of the student outcomes.

Table 5-2 Student outcomes and mapping to curricular and extra-curricular elements.

	Curriculum Component (credits)								Extra-curricular		
	VLPA/ IS	Commun	Math/ Statis	Science	CE core	Selective Elective	Capstone	Electives	Industry internships	Research project	TA Experience
(1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics			M	M	H	H	H	H	H	H	
(2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors					H	H	H	H	H	M	
(3) an ability to communicate effectively with a range of audiences		H			M	L	H		H	M	H
(4) an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgements, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts					M		H		H		
(5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives					L	L	H		H	M	H
(6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions					L	M	H	M	M	H	
(7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies					M	H	H	H	H	H	L

Table 5-3 Student outcomes and mapping to curriculum requirements

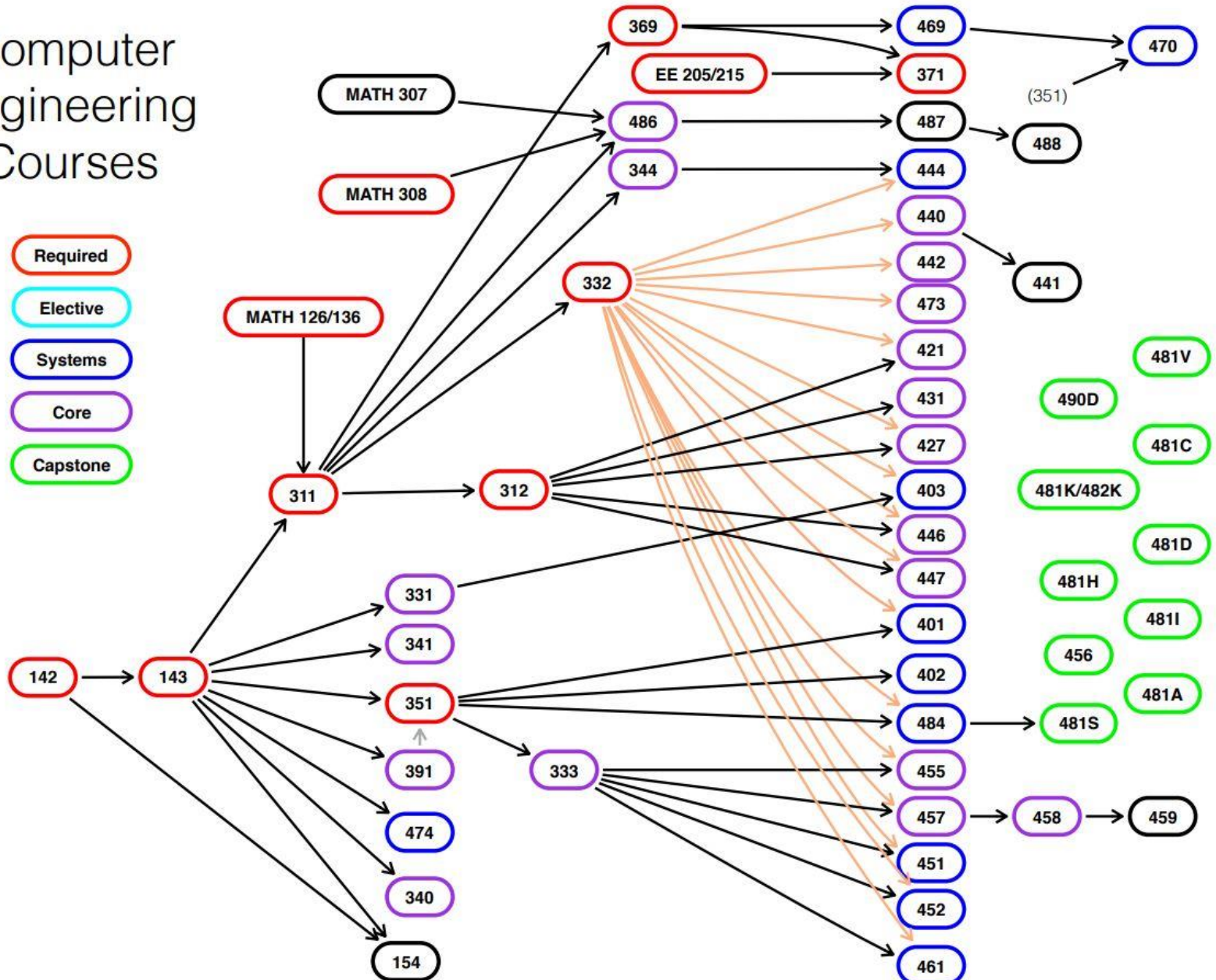
Outcome		1	2	3	4	5	6	7
CSE 142	Computer Prog I	H	H					M
CSE 143	Computer Prog II	H	H					
EE 205	Signal Conditioning	H		M	M	M	M	M
CSE 311	Foundations I	H					M	H
CSE 312	Foundations II	H		M			M	H
CSE 332	Data Abstractions	H	H				H	H
CSE 351	HW/SW Interface						H	H
CSE 369	Intro to Digital Design	H		M			H	H
EE 371	Digital Circuits and Systems	H	H				H	
CSE 401	Compilers	H				H	H	H
CSE 402	Domain Specific Languages							
CSE 403	Software Engineering	H	H	H	H	H	H	H
CSE 441	Adv. HCI	H	H	H	H	H	H	H
CSE 444	Databases	H		H			H	M
CSE 451	Operating Systems	H	H	L	M	H	H	H
CSE 452	Distributed Systems	H	H			H	H	
CSE 460	Animation Capstone	H	H	H	M	H	H	H
CSE 461	Networks	H	H			L	H	H
CSE 469	Comp. Arch I	H	M				M	H
CSE 470	Comp. Arch II	H	M				M	H
CSE 474	Intro to Embed. Sys.	H	M		H		M	
CSE 475	Embed Sys Capstone	H	H	H	H	H	H	H
EE 476	Digital Integ. Circuit Design	H	M	H	M	H	H	H
EE 477	VLSI II	H	H	H	H	H	H	H
CSE 481A	Op Sys Capstone	H	H	H	H	H	H	H

Outcome		1	2	3	4	5	6	7
CSE 481C	Robotics Capstone	H	H	H	H	H	H	H
CSE 481D	Games Capstone	H	H	H	H	H	H	H
CSE 481H	Accessibility Capstone	H	H	H	H	H	H	H
CSE 481I	Sound Capstone	H	H	H	H	H	H	H
CSE 481N	NLP Capstone	H	H	H	H	H	H	H
CSE 481S	Security Capstone	H	H	H	H	H	H	H
CSE 481V	VR Capstone	H	H	H	H	H	H	H
CSE 484	Security	H	H	H	H	H	H	H

A.4 Course Prerequisite Structure

Attach a flowchart or worksheet that illustrates the prerequisite structure of the program's required courses.

Computer Engineering Courses



A.5 and A.6 Meeting Degree Requirements

Describe how the program meets the requirements in terms of hours and depth of study for each subject area (Math and Basic Sciences, Engineering Topics) specifically addressed by either the general criteria or the program criteria.

Describe the broad education component and how it complements the technical content of the curriculum and how it is consistent with the program educational objectives.

The curriculum is structured to provide both depth and breadth. Students majoring in Computer Engineering have 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 virtual reality. 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 teamwork, 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 system electives
- 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.

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

Written and oral communication requirements include one technical writing course, and one general UW writing course in addition to their required English composition course. Mathematics includes one-year of calculus and a linear algebra class. We teach a required probability/statistics class as CSE 312. The science requirement includes two courses of Physics and 10 additional 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, bringing 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.

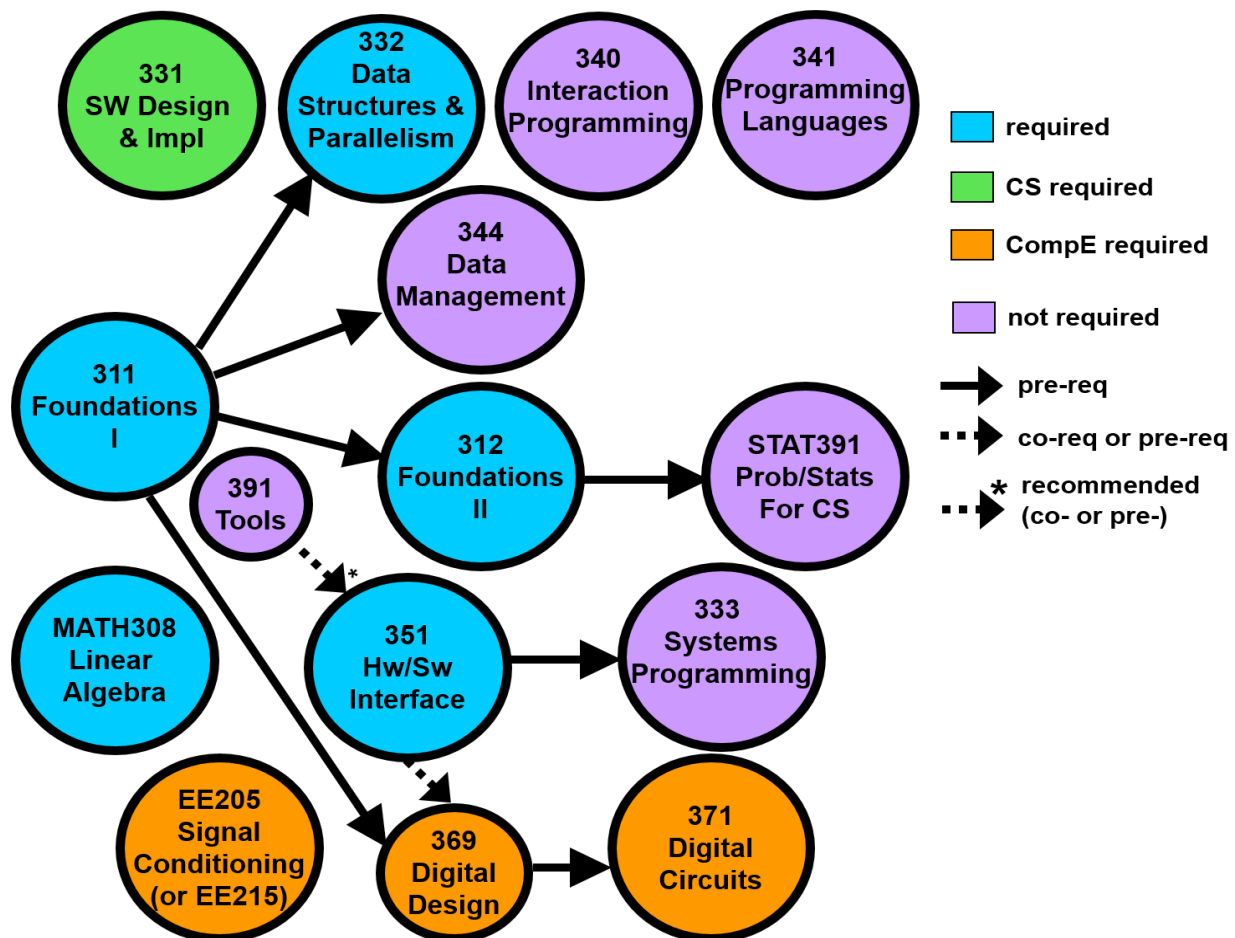


Figure 5-1 – Required Course Pre-requisite Structure

Figure 5-1 shows the pre-requisite structure of our 300-level courses. The heart of our curriculum 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 four courses: CSE 311, 312, 332, and 351. Computer Engineering students also take 12 additional foundation

credits: EE 205 or 215 (Introduction to Electrical Engineering or Introduction to Signal Conditioning) and CSE 369 (Digital Design) and CSE 371 (Digital Circuits).

The Computer Engineering curriculum is described in detail on the department undergraduate program site, www.cs.washington.edu/academics/ugrad/courses/requirements

A list of CSE core courses is at www.cs.washington.edu/academics/ugrad/curriculum/core

Systems Electives are at www.cs.washington.edu/academics/ugrad/curriculum/ce-systems

Senior electives are at www.cs.washington.edu/academics/ugrad/curriculum/electives

Capstone courses are at www.cs.washington.edu/students/ugrad/capstone/

Natural science courses are at www.cs.washington.edu/students/ugrad/natural_science/

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

Core requirements (36 credits): CSE 142, 143, 311, 312, 332, 351, 369, 371, and EE 215/205.

CE Senior Electives (36 credits)

- One course chosen from CSE 403, CSE/EE 474, or CSE 484.
- Three additional courses chosen from the Computer Engineering Systems Electives
- Two additional courses from the CSE Core Course list on the CSE site:
- One design capstone course from the approved list on the CSE site
- 4 credits of College of Engineering courses from the CSE electives list
- Additional courses to bring the total of CSE electives to 36
- Additional Engineering credits to bring the total Engineering credits to 36

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 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 (13 credits): English Composition, ENGR 231, 5 credit UW Writing course

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

A.7 Design Experience

Describe the major design experience that prepares students for engineering practice. Describe how this experience is based upon the knowledge and skills acquired in earlier coursework and incorporates appropriate engineering standards and multiple design constraints.

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 significant design component.

CSE401 – Introduction to Compiler Construction: This has always been a project 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 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.

CSE402 – Design and Implementation of Domain-Specific Language: This course uses a number of modern tools and concepts for implementing programming languages, typically by embedding new languages within higher-level languages. In the last 3 weeks of the course, students design and implement their own language using the ideas and techniques learned via earlier assignments. They not only gain the ability to succeed in building a new language, but in motivating the need for a special-purpose language and in designing how to meet that need.

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.

CSE 444 – Database System Internals: This course teaches the internals of relational database management systems and how to build such systems. It builds on the material in CSE 344. The course covers the full stack. Starting from a quick review of the relational model and SQL, the course dives into the architecture of a DBMS and then goes in depth into each component: storage manager, query executor, query optimizer, transactions, parallel processing, distributed transactions, data replication, architecture of NoSQL and NewSQL systems, etc. Students get to implement each of the components in their own Java DBMS. At the end of the quarter, successful students have a working DBMS that can execute SQL queries, update data, run concurrent transactions, recover from failures, and either execute in parallel or optimize queries.

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 452 - Distributed Systems: Distributed systems have become central to many aspects of how computers are used, from web applications to e-commerce to content distribution. This senior-level course covers abstractions and implementation techniques for the construction of distributed systems, including client server computing, the web, cloud computing, peer-to-peer systems, and distributed storage systems. The course has a demanding set of projects wherein students develop fault-tolerant distributed systems, including primary-backup systems, replicated state machines, and transactions across replicated state machines. The course project is designed to help students understand the implementation issues in modern cloud-scale systems, and they implement simplified versions of Google’s Chubby and Spanner systems.

CSE 461 – Introduction to Computer-Communications Networks: Students learn the fundamentals of networking layers, from physical through application. Additionally, they implement significant networked software that also highlights the operating system interfaces for networking and the behavior of real-world distributed software systems. There is considerable emphasis on dealing with errors and doing so efficiently. Project specifications are for protocols, some of which may be developed communally. Student implementations of the protocols are intended to inter-operate with each other and with reference implementations.

CSE474– Embedded Systems: The laboratories require the student to assess and analyze the assignment, then apply basic engineering knowledge to either solve the problem or state why (based upon their analysis) they are unable to fully satisfy the requirements. The laboratories assign a particular design problem to be solved. The final project brings all of the concepts together through the development of a (simplified) real world project. Each of the design projects provides a high-level requirements specification and a design specification for the problem that must be solved and each student must think about the broader implications of their designs. For the final project, the students are given only a set of requirements and they must interpret the requirements and implement the design specification.

CSE 469 – Computer Architecture I: Strong knowledge of hardware design and Verilog from EE 271 or CSE 369 is essential. The major goals of the class are to familiarize you with basic structure of microprocessors. As part of this, students will develop a Verilog implementation of a simple RISC microprocessor based upon the ARM instruction set. We will cover:

- Introduction to processor architecture.
- Assembly language programming.
- Computer Arithmetic.
- Performance measures.
- Processor Datapaths & Control.
- Pipelining.
- Memory hierarchy, caches, virtual memory.
- Advanced topics in computer architecture

CSE 470 – Computer Architecture II: Students learn key architectural techniques from original research papers and discuss in class how those solutions are implemented, their trade-offs and relevance to today. Students discuss and learn about contemporary needs and how key hardware design techniques are useful for addressing them. Students produce a written deep-dive summary of a research topic that synthesizes core knowledge in the area.

CSE 484 – Computer Security: Computer security and privacy are increasingly critical issues as our world becomes more computerized and interconnected. This senior-level course complements the rest of students' computer science education by providing a broad foundation in computer security and privacy, teaching them a "security mindset" and techniques for evaluating the designs of computer systems and ultimately designing systems

more resilient to potential attack. In this course, students study a broad range of topics including: threat modeling, software security, cryptography, malware, web security, web privacy, smartphone security, authentication, usable security, anonymity, physical security, and security for emerging technologies. Students must have maturity in both the mathematics of computer science (for the cryptography components of the course) and in the engineering of computer systems (for the labs and other course topics). The course includes challenging labs in which students are asked to find and exploit vulnerabilities in computing systems (including buffer overflows in C programs, vulnerabilities in web applications, and vulnerabilities in smart home systems), and to reflect on designs for mitigating these vulnerabilities. The course also includes a final project in which students independently explore in depth a computer security or privacy related topic of their choice, including consideration of relevant technical, legal, and ethical issues.

EE 476 - Digital Integrated Circuit Design: Students team up in groups of 3 to solve two design problems. One that centers around the design of a register file (full-custom layout), and another that varies as determined by the instructor. An example of a final project is the design of a custom 8-tap FIR filter (schematic only). The teams are judged on correctness, validation methodology, quality of design, speed and energy-delay product. Each group's project serves as an entry into the proposed Micron VLSI design contest. The contest will be judged by two engineers from Micron with (optional) consultation with the instructor. A cash prize of \$500 is proposed for the winning team of 3 at the final judging event. A \$300 second place prize can be awarded. It is recommended that the engineers make a 30-45m presentation on a Micron/Engineering related topic that is mutually agreed upon to be suitable for the audience. Given that the class is offered in Fall, Micron may choose to set aside an internship position for the winners of the VLSI design contest for the following summer.

EE 477 – VLSI II: Much of the class is heavily based on application of math, science, and engineering knowledge. This is emphasized in class and assessed through application in project hand calculations and planning. For each of the design projects, the student must analyze the requirements, then design, implement, and test the design, to verify its performance and characteristics. Students are encouraged to choose and execute projects to optimize their impact on society, and to estimate and evaluate this impact. Moreover, in modern ASIC design flows, there is an extensive requirements to perform analysis, synthesis, and to create explicit specifications and constraints. We discuss the tradeoffs in different engineering approaches and the greater societal consequences of developing VLSI systems. Students are required to work in teams to achieve their project goals. Teamwork is assessed in project work and in 'peer points'. Modern CAD tools require extensive use of engineering judgement and experimentation in order to attain high QoR. The projects push students to build these skills. The project requires that students choose a new application to target, which requires that they learn independently the required material. Lecture material continually emphasizes that today's technology is transitory and that the student must learn the basics so that these may form a foundation upon which they will understand and build future technologies. The need to continually augment one's education is emphasized. Recent journal and conference papers and new developments in digital VLSI are brought into class.

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.

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 approach, 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 embedded systems, virtual reality, operating systems, sound and media, animation, robotics, security, computational biology, human computer interaction, accessibility, natural language processing, and technology for resource constrained environments.

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 <https://www.youtube.com/user/uwcse>.

As an example of the scope, organization, and activities in our capstones is seen in the course page for the robotics capstone from Winter 2019: <https://sites.google.com/view/cse481wi>. We chose this capstone simply because it has the pertinent information on a single web page, making it easy to review.

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.

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 cataloguing the prokaryotic regulatory elements.

CSE 441 – Advanced HCI: Students will work in groups of three or four on a single project that parallels the experience of delivering an interactive prototype within a company or with a customer. Students are expected to already possess knowledge of appropriate HCI methods, and will focus on independently applying those methods in the context of your project. There will therefore be little lecture material in this course. Course staff will instead work closely with students to critique and advise on their group project. Students will experience the end-to-end product cycle from design to deployment.

CSE 460 – Animation: Students apply the knowledge gained from CSE 458 to produce an animated short film. Students will go through an animation industry standard production pipeline that involves modeling, shading, lighting, animating, rendering, and post-production.

CSE 475 – Embedded Systems 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 consist 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 481A – Operating Systems: 481A is a capstone course on operating systems, with an emphasis on virtualization. In lectures we introduce lvisor, a minimal x86 hypervisor based on KVM. We expect students to work on a project related to lvisor, finish exercises, discuss research papers, and make a presentation of their projects at the end of the quarter.

CSE 481C – Robotics Capstone: The robotics capstone teaches Computer Science & Engineering students the basics of robotics and gives them experience programming a mobile manipulator robot. The course covers robot motion, navigation, perception, manipulation, and user interaction through mini-lectures, labs and assignments. Students integrate these components to create autonomous or semi-autonomous robotic functionalities for a realistic robotic application. They learn to use libraries and tools within the most popular robot programming framework called ROS (Robot Operating System). The project gives students team-work experience with large scale software integration and gets them to explore opportunities for using robots to make people's lives easier.

CSE 481N - Natural Language Processing: In this capstone, students will work in small teams of 2-3 to build applications, prototype systems, and investigate scientific hypotheses using state of the art Natural Language Processing (NLP) technology. Areas of applications include text classification, information extraction, social media analysis, summarization, conversation (usually called dialogue by researchers), interpretation of deep neural models, question answering, and semantic parsing. In addition, cross-disciplinary applications are also encouraged, for example, image captioning, code generation from natural language descriptions, language based robot manipulation and navigation, connecting NLP with various disciplines such as computer vision, robotics, HCI, and programming languages.

CSE481S – Security: Student teams will be tasked with creating a computer security themed product. The work will progress from product conception to requirements to design to implementation to evaluation. Along the way, students will incorporate key computer security tools and practices, including threat modeling, penetration testing, and bug fixing. Examples include password managers, censorship resistance systems, and mobile payment systems.

CSE481V - AR/VR: Virtual and Augmented reality are promising technologies that are certain to make an impact on the future of business and entertainment. In this capstone, students will work in small project teams to build applications and prototype systems using state of the art Virtual Reality (VR) and Augmented Reality (AR) technology. Seattle is a nexus of VR tech, with Oculus Research, Valve, Microsoft (hololens), Google (cardboard, jump), and teams in the area. We will be developing on the latest VR/AR headsets and platforms, and will bring in leading VR experts for lectures and to supervise student projects. Students will experience the end-to-end product cycle from design to deployment, and learn about VR/AR technology and applications. The capstone culminates in a highly anticipated demo day where the students demonstrate their creations to other students, faculty and industry luminaries.

CSE 482 – Accessibility Capstone: Accessibility is quickly emerging as a leading consideration for product design and engineering. Disability is part of the human condition – almost everyone will be temporarily or permanently impaired at some point in life, and those who survive to old age will experience increasing difficulties. Disability is complex and heterogeneous, and the technological interventions to accommodate different abilities are wide ranging and vary with context. Many familiar technologies like voice recognition, text-to-speech, and gaze detection were initially engineered to assist people with disabilities gain more access and increase participation in daily life. Students will work in interdisciplinary project teams that include community members with expertise on project needs. Groups will follow participatory design practices and apply design and engineering skills to create technology solutions that increase independence and improve quality of life for people of all abilities. Teams will complete one

end-to-end product iteration cycle: ideation, design, specification refinement, prototype and usability testing

A.8 Extra-Curricular Activities and Co-op

If the program allows cooperative education to satisfy curricular requirements specifically addressed by either the general or program criteria, describe the academic component of this experience and how it is evaluated by the faculty.

A.8.1 Extra-Curricular Activities

We 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. For example, more than 70% of our students complete industry internships. Roughly 20-30% of our students participate in research with faculty and graduate students. Many of our undergraduates also publish research papers with faculty.

Finally, approximately 50% of our most recent graduating class 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 and our undergraduate curriculum, 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 Director, 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.8.2 Co-op Credits

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

A.9 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. In addition, we will have copies of textbooks used in our courses.

B. Course Syllabi

In Appendix A of the Self-Study Report, include a syllabus for each course used to satisfy the mathematics, science, and discipline-specific requirements required by Criterion 5 or by any applicable program criteria.

CRITERION 6. FACULTY

A. Faculty Qualifications

Describe the qualifications of the faculty and how they are adequate to cover all the curricular areas of the program and also meet any applicable program criteria. This description should include the composition, size, credentials, and experience of the faculty. Complete Table 6-1. Include faculty resumes in Appendix B.

The department operates with minimal hierarchy. Faculty of all ranks have similar teaching loads; instructional faculty (lecturers) have twice the teaching load of tenure-track faculty but often teach double sections of the same course in a given term. Junior faculty have annual peer reviews of teaching and meetings with the Director to discuss activity level and performance in teaching, research, and service. Faculty reviews include a reflective statement from the faculty member outlining their assessment of the past year and goals for the future.

There are 69 tenured or tenure track faculty:

- 35 Full Professors (4 joint appointments)
- 25 Associate Professors (5 joint appointments)
- 9 Assistant Professors (0 joint appts)

In addition, we have a strong group of instructional faculty:

- 2 Principal Lecturers
- 3 Senior Lecturers
- 5 Lecturers

The CSE faculty covers a wide range of the discipline and, collectively, has a wide range of experience, 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 national and international levels.

CSE is widely regarded as among the top ten programs in the nation. Thirty-nine current CSE faculty members have won Presidential/NSF Young Investigator Awards or NSF CAREER Awards. Six faculty members are ONR Young Investigator Award recipients. Five current (plus five former) faculty have held NSF Presidential Faculty Fellow or Presidential Early Career (PECASE) Awards. Twenty five have held Sloan Research Fellowships. Among current and emeritus senior faculty are nine Fulbright recipients, three Guggenheim recipients, twenty Fellows of the ACM, fourteen Fellows of the IEEE, two Fellows of the International Association for Pattern Recognition, five Fellows of the Association for the Advancement of Artificial Intelligence, three Fellows of the American Association for the Advancement of Science, four Fellows of the American Academy of Arts & Sciences, three recipients of ACM or IEEE Field Awards, three MacArthur Fellows, four (plus one Adjunct) Allen Distinguished Investigators, and four (plus ten Adjunct/Affiliate) Members of the National Academies.

Five faculty members have received the University of Washington Distinguished Teaching Award, two have received the University of Washington Marsha L. Landolt Distinguished Graduate Mentor Award, two have received the University of Washington Outstanding Public Service Award, one has received the David B. Thorud Leadership Award, one was named the University of Washington Annual Faculty Lecturer, and three have received the College of Engineering Faculty Achievement Award.

Our faculty continually strives to improve its teaching performance. Curriculum quality is enhanced through faculty self-assessments of their teaching performance after each course, peer evaluations of teaching, 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.

All our courses are evaluated by students. The faculty takes full advantage of the resources the University and College have to offer. In particular, many of our faculty have had the College of Engineering's Center for Teaching and Learning visit classes. Several of our instructional faculty are part of a campus-wide initiative on evidence-based teaching practices.

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. National visibility is high, with several faculty members serving on National Science Foundation advisory boards, National Research Council study panels, Department of Defense research organizations, and more.

Table 6-1 Faculty Qualifications

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		33	33		L	M	L
Anderson, Ruth	Ph. D., Computer Science University of Washington, 2006	O	NTT	FT		19	14		M	L	N
Anderson, Tom	PhD, Computer Science Washington, 1991	P	T	FT	4	27	21		H	M	L
Balazinska, Magdalena	PhD, Computer Science MIT, 2006	P	T	FT		13	13		H	H	M
Beame, Paul	PhD, , Computer Science Toronto, 1987	P	T	FT		31	31		H	H	L
Bodik, Rastislav	PhD, University of Pittsburgh, 1999	P	T	FT		19	4		M	H	M
Bricker, Lauren	Ph.D. Computer Science UW, 1998	O	NTT	FT	22	14	2		L	M	L
Cakmak, Maya	PhD, Robotics, Georgia Institute of Technology, 2013	AST	TT	FT	1	6	6		L	M	L
Ceze, Luis	PhD, Computer Science UIllinois Urbana-Champaign, 2007	ASC	T	FT		12	12			H	
Cheung, Alvin	PhD, Computer Science, MIT, 2015	AST	TT	FT		4	4		L		
Choi, Yejin	PhD, Computer Science, Cornell, 2010	ASC	T	FT		9	5		M	H	H
Curless, Brian	PhD, Electrical Engineering Stanford, 1997	P	T	FT		19	19		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
Domingos, Pedro	PhD, Computer Science UC-Irvine, 1997	P	T	FT		22	20		L	H	L
Ernst, Michael	PhD, Computer Science Washington, 2000	P	T	FT		20	11			H	M
Farhadi, Ali	PhD, Computer Science U Illinois Urbana-Champaign, 2011	ASC	TT	FT		7	7		L	H	H
Fogarty, James	PhD, Computer Science Carnegie Mellon, 2006	P	T	FT		13	13			H	L
Fox, Dieter	PhD, Computer Science U Bonn, 1998	ASC	T	FT	4	19	19			H	M
Froehlich, Jon	PhD, Computer Science, U. of Washington, 2011	ASC	T	FT		8	2				
Gollakota, Shyamnath	PhD, EE & Computer Science, MIT, 2013	ASC	TT	FT		7	7			H	
Grossman, Daniel	PhD, Computer Science Cornell, 2003	P	T	FT		16	16		H	H	L
Guestrin, Carlos	PhD, Computer Science Stanford, 2003	P	T	FT	1	15	7			H	
Hannaneh Hajishirzi	PhD, Computer Science UIUC, 2011	AST	TT	FT		1	4		M	H	M
Heer, Jeffrey	PhD, Computer Science UC Berkeley, 2008	P	TT	FT	6	11	6		M	M	H
Heimerl, Kurtis	PhD, Computer Science UC Berkeley, 2013	AST	TT	FT	3	3	3		L	M	L
Hemingway, Bruce	AB, Music Indiana U, 1973	O	NTT	PT	32	19	17		L	L	L
Hsia, Justin	PhD, Computer Science UC Berkeley, 2013	O	NTT	FT	1	4	3		L	L	N

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
Jamieson, Kevin	PhD, Electrical and Computer Engineering University of Wisconsin - Madison, 2015	AST	TT	FT	0	2	2		L	H	M
Just, René	PhD, Computer Science, Ulm University, 2013	AST	TT	FT	0	4	1		M	M	L
Karlin, Anna	PhD, Computer Science, Stanford, 1987	P	T	FT	6	23	23		M	H	L
Kemelmacher, Ira	PhD, Computer Science & Appl Math Weizmann Inst. Of Science, 2009	AST	TT	FT	1	7	7			H	
Kohno, Tadayoshi	PhD, Computer Science UC San Diego, 2006	P	T	FT		13	13			H	
Krishnamurthy, Arvind	PhD, Computer Science UC Berkeley, 1999	P	T	FT		20	14			H	
Ladner, Richard	PhD, Mathematics UC Berkeley, 1971	O	T	PT		45	48		H	H	M
Lazowska, Ed	PhD, Computer Science U Toronto, 1977	P	T	FT	1	41	41		H	M	H
Lee, James R	PhD, Computer Science UC Berkeley, 2005	P	T	FT	1	13	13		M	H	M
Lee, Su-In	PhD, Computational Biology Stanford, 2010	ASC	TT	Joint FT		11	10			H	
Lee, Yin Tat	PhD, Mathematics MIT, 2016	AST	TT	FT		2	2		L	H	M
Levy, Henry M	MS, Computer Science Washington, 1981	P	T	FT	8	36	36		H	L	L
Lin, Huijia	PhD, Computer Science Cornell, 2012	ASC	T	T		6	1		L	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
Mankoff, Jennifer	PhD Computer Science Georgia Tech, 2001	P	T	FT	18	18	2		H	L	L
Mahajan, Ratul	PhD Computer Science UW 2005	ASC	TT	FT	11	0.5	0.5		H	L	H
Mones, Barbara	MFA, Sculpture Rhode Island Schl of Design, 1979	O	NTT	FT	3	33	20		H	L	M
Oh, Seewong	PhD, Electrical Engineering Stafor, 2011	ASC	T	FT	2	12	12		H	H	M
Oskin, Mark	PhD, Computer Science UC Davis, 2001	P	T	FT	2	19	19		H	H	M
Oveis Gharan, Shayan	PhD, Computer Science MIT 2009	AST	TT	FT		4	4			H	
Patel, Shwetak	PhD, Computer Science Georgia Tech, 2008	P	TT	Joint FT	4	11	11		H	H	H
Perkins, Hal	MS, Computer Science Cornell, 1982	I	NTT	FT	2	37	20		L	L	N
Popovic, Zoran	PhD, Computer Science Carnegie Mellon, 1999	P	T	FT		20	20		N	H	N
Rao, Anup	PhD, Computer Science U Texas, Austin, 2007	AST	TT	FT		10	10			H	
Rao, Rajesh	PhD, Computer Science U Rochester, 1998	P	T	FT	2	19	19		M	H	L
Reges, Stuart	MS, Computer Science Stanford, 1982	O	NTT	FT	0	37	15		M	L	L
Reinecke, Katharina	PhD, Computer Science U Zurich, 2010	AST	TT	FT		5	3		M	L	L
Roesner, Franziska	PhD, Computer Science & Eng. UW, 2016	AST	TT	FT		5	5		M	M	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
Rothvoss, Thomas	PhD, Mathematics EPFL, 2009	ASC	T	Joint FT		5	5		M	M	N
Ruzzo, Larry	PhD, Computer Science UC Berkeley, 1978	P	T	FT	5	42	42		N	H	N
Schafer, Hunter	MS, Computer Science UW, 2018	O	NTT	FT		1	1		N	L	N
Seelig, Georg	PhD, Physics U of Geneva, 2003	ASC	T	Joint FT			10			H	
Seitz, Steve	PhD, Computer Science U Wisconsin, 1997	P	T	FT	3	22	19		L	H	L
Shapiro, Linda	PhD, Computer Science U Iowa, 1974	P	T	FT Joint	2	43	33		M	H	L
Smith, Joshua	PhD, Media Arts & Sciences MIT, 1999	P	TT	FT- Joint	11	9	9		L	H	M
Smith, Noah	PhD, Computer Science Johns Hopkins, 2006	P	T	FT	1	13	4		H	H	H
Suciu, Dan	PhD, Computer Science U Pennsylvania, 1995	P	T	FT	5	22	19		M	H	M
Tanimoto, Steve	PhD, Computer Science Princeton, 1975	P	T	FT	0	44	42		M	H	H
Tatlock, Zachary	PhD, Computer Science & Eng UC San Diego, 2014	AST	TT	FT		6	6		M	M	L
Taylor, Michael	PhD, EECS MIT, 2007	ASC	T	FT	2	13	2		M	H	L
Tessaro, Stefano	PhD, Computer Science ETCH Zurich, 2010	ASC	T	T		6	1		L	H	L
Tompa, Martin	PhD, Computer Science U Toronto, 1978	P	T	PT	4	37	37		L	H	H

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
Torlak, Emina	PhD, Computer Science MIT, 2009	ASC	T	FT	3	5	5		M	M	L
Wang, Xi	PhD, Computer Science MIT, 2014	AST	TT	FT		5	5				
Weld, Daniel	PhD, Artificial Intelligence MIT, 1988	P	T	FT		31	31		H	H	H
Zahorjan, John	PhD, Computer Science U Toronto, 1980	P	T	FT		39	39		L	L	L
Zettlemoyer, Luke	PhD, Computer Science MIT, 2009	ASC	T	FT		8	8		M	H	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

Complete Table 6-2, Faculty Workload Summary and describe this information in terms of workload expectations or requirements.

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	18:au: CSE 490b, 19wi: CSE 421, 19sp: CSE 482b	33%	50	17%	100%
Anderson, Ruth	FT	18au: CSE 332, CSE590E; 19wi: CSE 332, CSE590E; 19sp: CSE 351, CSE590E	80%	10%	10%	100%
Anderson, Tom	FT	sabbatical	22%	61%	17%	100%
Balazinska, Magdalena	FT	18au: CSED516; 19wi & 19sp: leave	22%	61%	17%	100%
Beame, Paul	FT	sabbatical	22%	61%	17%	100%
Bodik, Rastislav	FT	18au: release; 19wi: CSE501; 19sp: CSE402	33%	34%	33%	100%
Bricker, Lauren	FT	18au: CSE 190Z, CSE 154; 19wi: CSE 190Z; 19sp: CSE 190Z, CSE 154	50%	10%	40%	100%
Cakmak, Maya	FT	18au: MSTI 510 19wi: CSE 481 C	22%	61%	17%	100%
Ceze, Luis	FT		22%	61%	17%	100%
Cheung, Alvin	FT	Partial leave		80%	20%	
Choi, Yejin	FT	18sp: 481 N, 18wi: CSE 447, 19wi: CSE 517	22%	61%	17%	100%
Curless, Brian	FT	19sp: CSE P 557	11%	56%	33%	100%
Domingos, Pedro	FT				100%	100%
Ernst, Michael	FT		22%	61%	17%	100%
Farhadi, Ali	FT	18: au: CSE 599	10%	80%	10%	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	
Fogarty, James	FT	18au: CSE 510; 19wi: CSE440; 19sp: Release	22%	61%	17%	100%
Fox, Dieter	FT		11%	72%	17%	100%
Froehlich, Jon	FT	18wi, 19wi: HCID521; 18sp: CSE590; 19sp: CSE599	22%	61%	17%	100%
Gollakota, Shyamnath	FT	18wi, 19wi: HCID521 18sp: CSE590 19sp: CSE599	11%	72%	17%	100%
Grossman, Daniel	FT	18au: CSE341, 19wi: CSE390L; 19sp: CSE341	33%	25%	42%	100%
Hannaneh Hajishirzi	FT	19wi: CSE 573, 19sp: CSE 599	20%	80%		100%
Heer, Jeffrey	FT	19wi: CSE 573, 19sp: CSE 599	22%	61%	17%	100%
Heimerl, Kurtis	FT	17wi, 18wi, 18au: CSE 461	22%	61%	17%	100%
Hemmingway, Bruce	FT	18au: CSE/EE 475 ; 19wi: CSE 481i ; 19sp: CSE 131	33%	50%	17%	100%
Hsia, Justin	FT	18au: CSE 351 (4); 19wi: CSE 120 (5), CSE 369 (2); 19sp: CSE 333 (4), CSE 369 (2)	70%	5%	25%	100%
Jamieson, Kevin	FT	18au: CSE 546; 19sp: CSE 446	20%	75%	5%	100%
Just, René	FT	19wi: CSE 599; 19sp: CSEP 590	22%	61%	17%	100%
Karlin, Anna	FT	18au: CSE 312 ; 19wi: CSE 521p; 19sp: CSE 446	22%	61%	17%	100%
Kemelmacher, Ira	FT	19sp: CSE 481v	11%	72%	17%	100%
Kohno, Tadayoshi	FT	18au: CSE 484, 19wi: CSE 481S	11%	72%	17%	100%
Krishnamurthy, Arvind	FT	18au: CSE 550; 19sp: CSE 452	22%	61%	17%	100%
Ladner, Richard	PT			10%	30%	40%
Lazowska, Ed	FT	19wi: CSE 390L // 19wi: CSE 490A/599A1/P590 ENTRE 532A/532B	20%	20%	60%	100%
Lee, James R	FT	19wi: CSE 490z1; 19sp: CSE 525	30%	70%		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	
Lee, Su-In	FT		11%	72%	17%	100%
Lee, Yin Tat	FT	18au: CSE 421; 19wi: CSE 535	22%	61%	17%	100%
Levy, Henry M	FT		5%	35%	60%	100%
Lin, Huijia	FT	CSE 526	30%	40%	30%	100%
Mahajan, Ratul	FT	19wi: CSE 461	11%	72%	17%	100%
Mankoff, Jen	FT	19F: CSE340prep; CSE590W; 19W CSE590W; 19sp: CSE340; CSE590W	50%	50%	20%	120%
Mones, Barbara	FT		44%	39%	17%	100%
Oh, Seewong	FT	19sp: CSE/STAT416	27%	63%	10%	100%
Oskin, Mark	FT		33%	50%	17%	100%
Oveis Gharan, Shayan	FT	18au: CSE 521; 19sp: CSE 421	22%	61%	17%	100%
Patel, Shwetak	FT		11%	60%	29%	100%
Perkins, Hal	FT	18au: CSE 333, CSE 401; 19wi: CSE 333, CSE 413; 19sp: CSE331 (x2); 19su: CSE 331	90%		10%	100%
Popovic, Zoran	FT				100%	100%
Rao, Anup	FT		22%	61%	17%	100%
Rao, Rajesh	FT	18au: CSE 490N	22%	61%	17%	100%
Reges, Stuart	FT	18au: CSE 143X, CSE390H' 19wi: CSE142, CSE390H; 19sp: CSE143, CSE390H	90%		10%	100%
Reinecke, Katharina	FT	18au: CSE 440; 18sp: CSE 441; 19sp: CSE 441	22%	61%	17%	100%
Roesner, Franziska	FT	18au: CSE 599B, CSE 590Y; 19wi: CSE 564, CSE 590Y; 19sp: CSE 484 / 584M	22%	61%	17%	100%
Rothvoss, Thomas	FT	19wi: CSE 583; 19sp: CSE 311	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	
Ruzzo, Larry	FT	18au: CSE 427, CSE 590C; 19wi: CSE 417, CSE 590C; 19sp: CSE 427, CSE 590C	33%	50%	17%	100%
Schafer, Hunter	FT	18au: CSE 143; 19wi: CSE 143; 19sp: CSE 163	90%		10%	100%
Seelig, Georg	FT		11%	72%	17%	100%
Seitz, Steve	FT		11%	72%	17%	100%
Shapiro, Linda	FT	18au: EE 562; 19wi: CSE 473; 19sp: CSE/EE 576	25%	50%	25%	100%
Smith, Joshua	FT			100%		100%
Smith, Noah	FT	19wi: CSE 447; 19sp: CSE 481N	25%	50%	25%	100%
Suciu, Dan	FT	18au: CSE344; 19wi: CSE344; 19Sp: CSE414	33%	50%	17%	100%
Tanimoto, Steve	FT	18au CSE 473, 19wi CSE 415, 19sp CSE 415	22%	61%	17%	100%
Tatlock, Zachary	FT	18au CSE 505, 19wi CSE 341, 19sp CSE P505	33%	50%	17%	100%
Taylor, Michael	FT	19wi CSE 567, EE 477, EE 525	20%	63%	17%	100%
Tessaro, Stefano	FT	CSE P590	30%	40%	30%	100%
Tompa, Martin	PT	19wi CSE 312	100%			100%
Torlak, Emina	FT		22%	61%	17%	100%
Wang, Xi	FT		22%	61%	17%	100%
Weld, Daniel	FT	CSEP 573	12%	30%	8%	50%
Wortzman, Brett	FT	18au: CSE 142, CSE 391; 19wi: CSE 391, CSE 390 HA; 19sp: CSE 142, CSE 391, CSE 390 HA	80%	5%	15%	100%
Zahorjan, John	FT	18au CSE 481SYS ; 19wi CAE 451; 19sp CSE 461	60%	10%	30%	100%
Zettlemoyer, Luke	FT	18au CSEP 517; 19sp CSE 473	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

Discuss the adequacy of the size of the faculty and describe the extent and quality of faculty involvement in interactions with students, student advising and counseling, university service activities, professional development, and interactions with industrial and professional practitioners including employers of students.

The department has about 75 tenure-line faculty members (counting joint appointments fractionally), 10 instructors, and around 35 postdocs.

Since 2013 the department has made 27 tenure-track hires, a central piece of our aggressive growth. The most relevant for CompE are Rene Just (403), Ratul Mahajan (461), Michael Taylor (VLSI), Ras Bodik (402), Kurtis Heimerl (461, ICTD), and Franzi Roesner (484).

D. Professional Development

Provide detailed descriptions of professional development activities for each faculty member.

In CSE, sabbaticals are permitted every 7 years and are supported by providing 2/3's salary over three quarters. The department has very liberal policies with regard to faculty leaves to spend time in industry or to spin out department-developed technology. In general, we support faculty members' requests for industrial interaction as 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

Describe the role played by the faculty with respect to course creation, modification, and evaluation, their role in the definition and revision of program educational objectives and student outcomes, and their role in the attainment of the student outcomes. Describe the roles of others on campus, e.g., dean or provost, with respect to these areas.

Faculty take the lead in delivering existing courses, proposing new courses, evaluating individual courses and the curriculum as a whole, and in assigning grades. While we provide similar experiences within a course from quarter to quarter, there is considerable flexibility at the upper level, which allows faculty to innovate rapidly on assignments and, to a lesser extent, on topics.

Substantial curriculum revisions, like the introduction of courses and changes to graduation requirements, must pass through our Curriculum Committee, then through our faculty, and finally are presented to the College and University for approval.

Program educational objectives are defined by the faculty, after review by the Curriculum Committee and our Executive Committee.

CRITERION 7. FACILITIES

A. Offices, Classrooms and Laboratories

Summarize each of the program's facilities in terms of their ability to support the attainment of the student outcomes and to provide an atmosphere conducive to learning.

1. [Offices \(such as administrative, faculty, clerical, and teaching assistants\) and any associated equipment that is typically available there.](#)

With the opening of the Gates Center, the Allen School now has 100 faculty offices (55 in the Allen Center and 45 in the Gates Center), and 137 staff offices (administrative, advising, post-doctoral researchers, technical staff, and project research technical staff; 90 in the Allen Center and 47 in the Gates Center).

2. [Classrooms and associated equipment that are typically available where the program courses are taught.](#)

Our new Gates Center facility includes a state of the art, tiered 240-person lecture hall for large classes and Allen School colloquia including sophisticated video recording and streaming infrastructure. The Gates Center also has two large 100 person classrooms for a total of 10,000 sq feet of instructional classroom space. The Allen Center does not have any classrooms. Beyond those three classrooms located within the Gates Center, CSE leverages many classrooms across campus for our courses.

3. [Laboratory facilities including those containing computers \(describe available hardware and software\) and the associated tools and equipment that support instruction. Include those facilities used by students in the program even if they are not dedicated to the program and state the times they are available to students. Complete Appendix C containing a listing of the major pieces of equipment used by the program in support of instruction.](#)

The Allen School maintains a number of computing enabled laboratories, with managed workstations (Windows and Linux OS platforms), for both general computing purposes as well as for course-specific purposes. We have six general access computing labs, with approx. 140 PCs, and seating for 350 students. These workstations are Intel based, were acquired within the past four years, and are provided with dual-displays. We also provide a limited number of GPU

enabled workstations. In these labs we also provide table top power receptacles and stand-alone displays to accommodate personal devices (laptops, etc). University WiFi is available throughout our instructional spaces, as well as in the public areas between the buildings. We have seven ‘special purpose’ labs with capacity for 90 workstations and seating for as many as 170 students.

General access labs are available to full-time CSE students for any use that furthers student educational goals.

Special purpose labs are limited to use by students enrolled in specific courses. Examples of capstone courses include; Accessibility, Animation Computer Security, Software Design, Game Design, Virtual Reality, and multiple Robotics offerings.

The following computing 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 brief interruptions of weekly maintenance, course restriction, or existing reservations.

Allen Center General Computing Labs [CSE 002, 003, 006]
80 Workstations, Seating capacity 210 students

Allen Hardware Design/Embedded Systems Capstone Lab [CSE003E]
15 Workstations, Seating capacity for 30 students

Allen Center Special Projects Lab [CSE 003D]
HW varies by quarter, seating capacity for ~20 students

Allen Center Capstone Lab [CSE022]
HW varies by quarter, seating capacity for ~40 students

Gates Center General Computing Labs [CSE2 110, 124, 129]
60 Workstations, Seating capacity 140 students

Gates Center Animation Capstone Labs [CSE2 141, 143]
24 Workstations, Seating capacity 50 students

Gates Center Capstone Labs [CSE2 128, 130]
HW varies by quarter, seating capacity for ~60 students

A list of our supported/installed software packages for both platforms are available at the following links:

Windows software list: <https://www.cs.washington.edu/lab/facilities/instr-sw-win>
Linux package list: <https://www.cs.washington.edu/lab/linux/instructional/software>

Our new Gates Center facility includes an undergraduate commons where students can study, relax and work together, with adjacent computing labs, TA meeting rooms and collaboration spaces. Four workrooms for our outstanding interdisciplinary computer animation capstone, and other capstone design courses. Improved facilities for teaching assistants to meet with students, as well as a large advising suite where all undergraduate and graduate advisors are located. Interview rooms where industry representatives can meet with students. A total of more than 13,000 sq feet dedicated to undergraduate labs and support spaces.

The Hardware Design/Embedded Systems Capstone Lab (CSE 003E) additionally contains many test instruments and hardware platforms for design of hardware projects and embedded systems. The lab has 16 workbenches, equipped with a workstation, oscilloscope, logic analyzer, function generator and power supply. Additional test equipment like multimeters and logic programmers are also available.

B. Computing Resources

Describe any computing resources (workstations, servers, storage, networks including software) in addition to those described in the laboratories in Part A, which are used by the students in the program. Include a discussion of the accessibility of university-wide computing resources available to all students via various locations such as student housing, library, student union, off-campus, etc. State the hours the various computing facilities are open to students. Assess the adequacy of these facilities to support the scholarly and professional activities of the students and faculty in the program.

The Allen School provides a substantial instructional Linux compute/cycle cluster consisting of 8 servers where course software is installed. We also provide a Windows Virtual Lab (VDI) with 18 nodes that can be connected to by personal devices. There are also dedicated Linux and Windows servers available for special-purpose projects. We provide students with access to local high-bandwidth file storage, and hosting for both personal and class project oriented web pages. We provide hosting and support for course database projects - primarily PostgreSQL and MySQL/MariaDB. Secure Git repository management, code reviews, issue tracking, wikis are provided via our locally-hosted GitLab service. We perform nightly incremental backups of data on these systems, then replicate that data to a secondary site, and can restore files by request or in case of catastrophic system failure. We maintain disaster recovery copies of most research data on 'off-site' central IT tape service.

These resources are available 24 hours a day, 7 days a week, to all CSE students and others enrolled in CSE major and graduate courses.

Other University campus computing labs are also available to CSE students. The University Libraries offers some of the largest labs on campus with hundreds of PCs. Most library labs are open to students 24 hours a day M-F during the quarter and more limited hours on weekends. More recently many of the development tools required by CSE courses are also distributed on these campus computers.

C. Guidance

Describe how students in the program are provided appropriate guidance regarding the use of the tools, equipment, computing resources, and laboratories.

Use of computing is inherent in all our courses, and students acquire needed skills as part of doing course work. Our tech staff team provides general help. A help desk is open every day, and additional help is available online. Students receive an introduction and overview of the Computing lab facilities, IT services and policies during orientation to the School. The School maintains extensive webpages that outline these IT services <https://www.cs.washington.edu/lab> and also provides an overview and links to UW and School 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. Additionally, the technical group sends out a periodic newsletter announcing new services or reminders about existing services. 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

Describe the policies and procedures for maintaining and upgrading the tools, equipment, computing resources, and laboratories used by students and faculty in the program.

The Allen School has a full-time IT and Facilities teams dedicated to the care and upgrading of all the facilities mentioned above. Most computing hardware is on a planned 5 year refresh cycle. Student assistants on both the IT and Facilities team check laboratories daily. Instructional computing equipment is monitored by an hourly ‘ping’ utility that gauges relative health workstations 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 (often daily) to ensure they are in working order and that supplies are stocked. 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. The School has written many successful grants to a local Student Tech Fee Program (uwstf.org) every few years to provide funding for new computing equipment. The Allen School provides additional funding from time to time to purchase new computing equipment for use in instruction. Computing hardware and facility upgrade opportunities are explored at least once a year.

E. Library Services

Describe and evaluate the capability of the library (or libraries) to serve the program including the adequacy of the library’s technical collection relative to the needs of the program and the faculty, the

adequacy of the process by which faculty may request the library to order books or subscriptions, the library's systems for locating and obtaining electronic information, and any other library services relevant to the needs of the program.

The UW College of Engineering Library is a great resource for students, 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

Describe how the program ensures the facilities, tools, and equipment used in the program are safe for their intended purposes. (See the 2019-2020 APPM section I.E.5.b.(1).)

Most of our computing 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 warning signage posted and/or required supervision by trained course staff if that equipment is to be operated by students.

CRITERION 8. INSTITUTIONAL SUPPORT

A. Leadership

Describe the leadership of the program and discuss its adequacy to ensure the quality and continuity of the program and how the leadership is involved in decisions that affect the program.

As of 2018-2019, the Allen School leadership consists of the Director, Deputy Director (includes educational oversight), and four Associate Directors (facilities, graduate studies, development & outreach, and research & innovation).

The Director is appointed by the Dean, typically for a five-year renewable term. Levy was appointed in 2006 and agreed to several extensions, but will step down in Summer 2019. The Associate Directors, including Deputy Director, are appointed by the Director and for duties that correspond with current departmental needs and individual interests.

The Allen School executive committee includes the leadership described above and five additional members elected by the full faculty to one-year 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., temporary instructors, 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 between the school’s leadership and its faculty.

B. Program Budget and Financial Support

- Describe the process used to establish the program’s budget and provide evidence of continuity of institutional support for the program. Include the sources of financial support including both permanent (recurring) and temporary (one-time) funds.
- Describe how teaching is supported by the institution in terms of graders, teaching assistants, teaching workshops, etc.
- To the extent not described above, describe how resources are provided to acquire, maintain, and upgrade the infrastructures, facilities, and equipment used in the program.
- Assess the adequacy of the resources described in this section with respect to the students in the program being able to attain the student outcomes.

The Allen School has a strong foundation of funding provided by the State of Washington. Since 2012, the state legislature has, in total, increased our funding by \$11M per year to enable nearly tripling the total number of undergraduate degrees (roughly 20% of which will be in computer engineering). The annual budget for the current fiscal year (FY19) including salaries and operations is roughly \$25 million, not including research expenditures of another nearly \$20M. The majority of our state funding supports personnel including faculty. State funding

supports 14 student services staff (across all degree programs), 25 technical staff, and administrative staff including the Assistant to the Director, External Relations Director, two Receptionists (for our two buildings), Facilities Manager, Events Coordinator, Administrator and 7 fiscal staff, grant/research staff, HR and payroll. State funding is augmented from several sources including CSE's industrial affiliates program, corporate and individual gifts, indirect cost returns (to the department) resulting from faculty awarded grants and contracts, and endowments. 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 and chairs, fellowships, and scholarships. CSE has 45 endowed scholarships, 15 endowed fellowships and 16 endowed professorships and chairs. Annually over 50 undergraduate scholarships are awarded. The School's discretionary endowment funds will going forward generate over \$2.18M annually in discretionary funds. Over 100 companies contribute to CSE's industrial Affiliates Program annually.

Over \$4M in state funding is used to provide 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 80 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

Describe the adequacy of the staff (administrative, instructional, and technical) and institutional services provided to the program. Discuss methods used to retain and train staff.

CSE has roughly 120 staff, with about 70 "core" Allen School staff and the other devoted to particular research projects or research centers. 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. These averages are quite high but in fact understate the case because our recent growth has included substantial staff expansion and new-hires drive down the average by definition. CSE is a supportive environment for staff, resulting in a high retention rate. Senior staff mentor and provide training for new staff hires. Staff are also encouraged to attend professional conferences and professional development opportunities provided by the University of Washington. Performance evaluations are regular, documented, and constructive. In 2016, we hired a full-time human-resources manager to help with our staff growth. Staff development through attending conferences and training classes is encouraged and supported, both financially and with release time.

D. Faculty Hiring and Retention

- Describe the process for hiring of new faculty.

- Describe strategies used to retain current qualified faculty

D.1: Hiring of new faculty

Faculty hiring is managed through a Faculty Recruiting Committee (FRC), appointed by the Director. FRC is responsible for writing and publishing advertisements for open positions, evaluating applicants in collaboration with faculty in relevant 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 throughout the recruiting season. Final offers are chosen through a vote of the entire faculty. Last year (2018) the department hired 9 new faculty members – 3 senior faculty members at the Associate Professor level, 4 junior faculty members at the Assistant Professor level, and 2 Lecturers. This year (2019), our search is not yet concluded at the time of this writing. We have made 9 Assistant Professor offers and 3 Lecturer offers. Currently 1 Assistant Professor offer has been accepted and 1 declined, while 1 Lecturer offer has been accepted and 2 declined. We will learn about the other 7 outstanding offers in the weeks ahead.

The School 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, Amazon, 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. Of particular note is a close relationship with the Allen Institute for Artificial Intelligence which performs open research to benefit humanity and is directed by a long-time faculty member.

Overall, since 2013, a five-year period, we have hired 35 total faculty while losing 8 tenure-track faculty (3 to industry, 2 to other institutions [Stanford, Berkeley], 2 to retirement, 1 to death) and 3 lecturers (2 to industry, 1 to another institution [Cal Tech]). At our size of 75 faculty FTE, we anticipate an average of 2 departures per year as a natural consequence of scale.

Proactive retention raises and other salary-increasing mechanisms (such as “A/B salary” – see below) have helped to increase salaries to remain competitive.

D.2 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 ensure 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. The University's policies on A/B salaries are in flux and this opportunity may not be available going forward.

E. Support of Faculty Professional Development

Describe the adequacy of support for faculty professional development, how such activities such as sabbaticals, travel, workshops, seminars, etc., are planned and supported.

At UW, 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. In a time of industry boom, particularly in our region, it is also essential for faculty retention.

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. In particular, the eScience Institute and the Design, Use, Build (DUB) grassroots organization for HCI are strong campus entities to which we have provided substantial leadership.

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:

- Center for Teaching & Learning (CTL)

CTL supports the University of Washington, College of Engineering's mission by taking a leadership role in developing and supporting engineering instructional excellence. The CTL 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 CTL approach to professional faculty development begins with meeting and resolving the immediate concerns of faculty members. Simultaneously CTL 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. CTL services are available to all faculty members in the College of Engineering. For more information on CTL services see the CTL description in Table D-4 *Non-academic Support Units*.

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 2017 Engineering by the Numbers reports UW has 24.2% women faculty in engineering compared to a national average of 17%. Further, out of the top 50 schools with engineering programs as ranked by US News & World Report, UW ranks second highest for percentage of women faculty in engineering.

Figure 8-1 shows the percentage of women faculty in Engineering, compared to the national average.

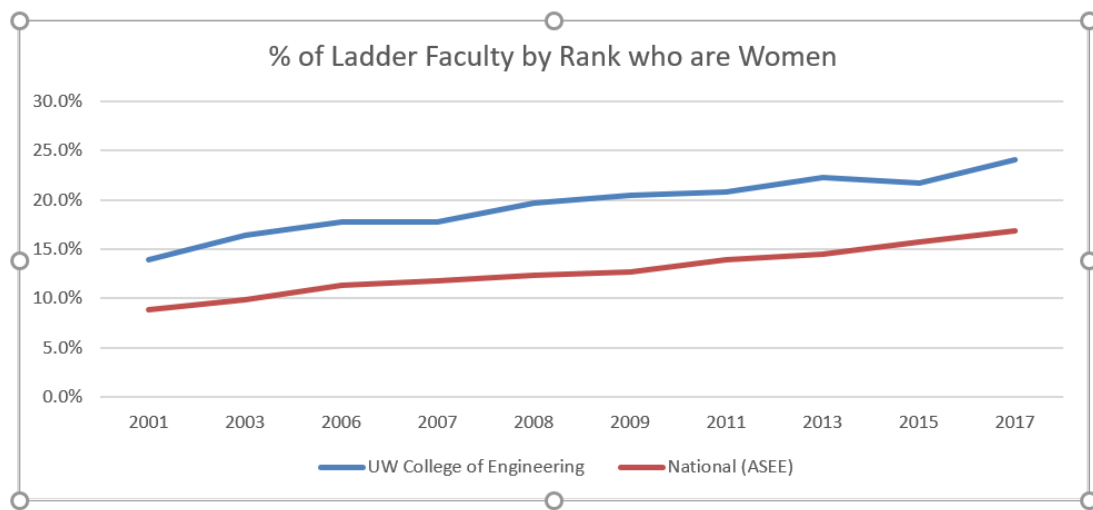


Figure 8-1 Percentage of Faculty Who Are Women

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 298 unique faculty members between 2003 and 2018.

- 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

Please see Section 5 on Curriculum.

APPENDICES

Appendix A – Course Syllabi

Written & Oral Communication

ENGR 231 – Introduction to Technical Communications

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

Textbook/Materials - M. Markel, Practical Strategies for Technical Communication, Second Edition. Boston, MA: Bedford/St. Martin's, 2016.

Catalog Description: This course introduces engineering undergraduates to the fundamental technical communication processes associated with writing, communication design, and oral presentations. Assignments focus on contemporary issues in engineering with a focus on ethics and sustainable engineering practices.

Pre-requisites:

Either both ENGL 109 and ENGL 110, or any one of the following: C LIT 240, ENGL 111, ENGL 121, ENGL 131, ENGL 182, ENGL 197, ENGL 198, ENGL 199, ENGL 281, ENGL 282, ENGL 297, ENGL 298, ENGL 299, ENGL 381, ENGL 382, HONORS 205, HONORS 345

Required: Yes

Coordination: College of Engineering

Learning Objectives/Content:

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

- Implement successful communication strategies based on a document's or presentation's audience, purpose, and context.
- Integrate text and visuals to convey complex, technical information.
- Revise documents for content, organization, and writing style.
- Using library research skills and knowledge of citation practices, conduct self-directed inquiry to identify, critically evaluate, and cite relevant literature.
- Provide feedback to others on their writing, speaking, and teamwork abilities.
- Demonstrate ability to work in teams and manage team projects.
- Design and deliver effective oral presentations.
- Understand ethics and sustainability in engineering.

ABET Outcomes:

- (4) An ability to communicate effectively with a range of audiences.

- (5) An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.
- (6) An ability to recognize the ongoing need to acquire new knowledge, to choose appropriate learning strategies, and to apply this knowledge.
- (7) An ability to function effectively as a member or leader of a team that establishes goals, plans tasks, meets deadlines, and creates a collaborative and inclusive environment.

Mathematics & Natural Sciences

CHEM 142 General Chemistry

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

Textbook:

Chemical Principles, 7th ed., Zumdahl/Decoste (custom-split Chem 142 version contains, in the order of coverage: Chapters 2, 12, 13, 3, 4, 15, & 5, as well as the Student Solutions Manual).

Catalog Description: For science and engineering majors. Atomic nature of matter, quantum mechanics, ionic and covalent bonding, molecular geometry, stoichiometry, solution stoichiometry, kinetics, and gas laws. Includes laboratory. Cannot be taken for credit if CHEM 120 already taken. Prerequisite: Either CHEM 110, or a passing score in the General Chemistry Placement exam, or a score of 1 on Chemistry AP test. Offered: AWSpS.

Required: No

Coordination: Department of Chemistry

Learning Objectives:

Students who successfully complete CHEM 142 will be able to

- Explain the chemical and physical behavior of matter based on the modern atomic theory, quantum mechanics, and the resulting atomic periodicity.
- Describe the formation and energetics of chemical bonds based on electrostatic forces.
- Describe and predict the structure of covalent and ionic compounds
- Describe the physical and chemical changes taking place in chemical reactions at both the particulate and macroscopic levels.
 - Write balanced chemical equations for acid-base, precipitation, and oxidation-reduction reactions and use the balanced equations to predict quantities of reactants and products.
- Explain the rate of chemical reactions and the conditions that influence the rate using rate laws, reaction mechanisms, and collision theory.
- Explain the behavior of gas phase chemical systems at the particulate and macroscopic level using ideal gas behavior.
- Develop skill in visualizing the particulate level as related to the concepts listed above. • Clearly define a problem and develop solutions for that problem including the use of central and auxiliary equations and conversion factors.
- Relate empirical observations, particularly in the laboratory portion of the course, to concepts listed above.

- Demonstrate laboratory, data analysis, and scientific writing skills.

CHEM 152 General Chemistry

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

Textbook, Supplemental Materials:

Chemical Principles, 7th ed., Zumdahl/Decoste (custom-split Chem 152 version contains, in the order of coverage: Chapters 6, 7, 8, 9, 10, & 11, as well as the Student Solutions Manual

Catalog Description: Gas phase and aqueous equilibria, thermochemistry, thermodynamics, and electrochemistry. Includes laboratory. No more than 5 credits can be counted toward graduation from the following course group: CHEM 152, CHEM 155. Prerequisite: minimum grade of 1.7 in either CHEM 142, CHEM 143 or CHEM 145. Offered: AWSpS.

Coordination: Department of Chemistry

Required: No

Learning Objectives/Course Topics

Students who successfully complete CHEM 162 will be able to: ☐

- Explain the properties of chemical molecules using bonding models, including hybridization and molecular orbital theory, with the understanding of their limitations. ☐
- At a beginning level, analyze spectroscopic results to determine the structure of molecules. ☐
- Use isomerism (structural, geometric, and stereo) to explain variation in chemical and physical properties. ☐
- Explain macroscopic properties based on intermolecular forces within the chemical system. ☐
- Describe the structure and properties of the liquid and solid states, as well as phase changes, at the particulate and macroscopic levels.
- Explain the chemical, physical, and thermodynamic properties of solutions at the particulate and macroscopic level. ☐
- Apply bonding models to the structural study of organic molecules and transition metal coordination complexes. ☐
- Illustrate the concepts of kinetics, thermodynamics, and equilibria through application to organic and transition metal chemistry.

- Develop skill in visualizing the particulate level as related to the concepts above. □
- Relate empirical observations, particularly in the laboratory portion of the course, to concepts listed above. □
- 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, 8th Edition, Early Transcendentals" by Stewart.

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: either a minimum grade of 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:

Tangents to circles Sec. 10.1 or supplement reading on parametric equations Sec. 2.1-Tangents and velocity Sec. 2.2-Limits
Sec 2.3-Calculating Limits Sec.2.5-Continuity Sec.2.6-Asymptotes
Sec. 2.7-Derivatives Sec. 2.8-Derivative Function Sec. 3.1-Derivative rules
Sec. 3.2-More derivative rules Sec. 3.3-Trig derivatives Sec. 3.4-Chain rule
Midterm Review Midterm #1 Sec. 3.4-More chain rule Sec. 10.2-Derivatives and parametrized curves
Sec. 3.5-Implicit differentiation Sec. 3.6-Logarithmic differentiation Sec. 3.9- Related rates
Sec. 3.9-More Related rates Sec. 3.10-Linear approximation Sec. 4.1-Basics on min and max values
Midterm Review Midterm #2 Sec. 4.1-More on min and max values Sec. 4.3- Derivatives and shape of a curve
Sec. 4.4-L'Hôpital's rule Sec. 4.5-Curve sketching Sec. 4.7-Optimization
Sec. 4.7-More Optimization Final Exam Review Final Exam

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:

Study Guide Exam Archive Topics and Textbook Sections
Antiderivatives, Areas and Distances, The Definite Integral
Areas between Curves, Volumes By Slicing, Volumes of Solids of Revolutions via the Disks/Washers method and via the Shells Method
Revolutions via the Disks/Washers method and via the Shells Method
Applications: More Volumes; Review. Midterm #1. Applications: Work
Applications: Work; Average Value of a Function;
Techniques of Integration: Integration by Parts, Products of Trig Functions
More Techniques: Trigonometric Substitution, Partial Fractions, Combining Methods
Approximation of Integrals; Improper Integrals; Applications: Arc Length of a Curve.
Applications: Center of Mass. Review. Midterm #2. Intro to Differential Equations
Diff Eqs: Solving Separable Equations, Various Applications
Archive More Diff Eqs

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: minimum grade of 2.0 in MATH 125, 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:

Intro	+ Section 12.1: 3D coordinate system Section 12.2: vectors Section 12.3: dot products, projections
Week 2	Section 12.4, 12.5: dot products, lines in space Section 12.5: planes in space Section 12.6: cylinders and quadric surfaces
Week 3	Section 13.1: vector functions and curves Section 13.2: derivatives and integrals of vector functions Section 13.3: curvature
Week 4	Section 13.3: normal and binormal vectors, normal plane Exam I Section 13.4: velocity, speed, acceleration
Week 5	Section 14.1: multi-variable functions Section 14.3, 14.4: partial derivatives, tangent planes Section 14.4, 14.7: differentials, optimization
Week 6	Section 14.7: Optimization Section 15.1: double integrals over rectangles

	Section 15.2: double integrals over general regions
Week 7	Section 10.3: polar coordinates Section 15.3: double integrals in polar coordinates Section 15.4: mass and center of mass
Week 8	Exam II Taylor Notes Section 1: first-order Taylor polynomials, including error bound Taylor Notes Section 2, 3: second- and higher-order Taylor polynomials, including error bound
Week 9	NO CLASS: Memorial Day Taylor Notes Section 3, 4: Taylor polynomials and Taylor series Taylor Notes Section 4, 5: new series from old
Week 10	Taylor Notes Section 5: more series Review for Final Exam

MATH 307 Introduction to Differential Equations

Credits: 3 Credits (3 hours lecture)

Text: Boyce and Diprima, Elementary Differential Equations and Boundary Value Problems or Elementary Differential Equations.

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

Prerequisite: Math 125; Math 126 strongly recommended.

Coordination: Department of Mathematics

Required: No

Topics Covered:

Some Basic Modelling § 1.1
Solutions to Differential Equations § 1.2
Direction Fields § 1.1
Separable First Order ODE's § 2.2
Linear First Order ODE's § 2.1
Modelling with First Order ODE's § 2.3
Population Dynamics § 2.5
Second Order Constant Coefficient ODE's § 3.1
Homogeneous equations with distinct real roots § 3.1
Complex numbers and Euler's formula -- notes on complex numbers
Homogeneous equations with complex roots § 3.3

MATH 308 Matrix Algebra with Applications

Credits: 3 Credits (3 lecture contact hours)

Text: Linear Algebra with Applications, Second Edition, by Jeffrey Holt.

Catalog Description: Systems of linear equations, vector spaces, matrices, subspaces, orthogonality, least squares, eigenvalues, eigenvectors, applications. For students in engineering, mathematics, and the sciences. Prerequisite: minimum grade of 2.0 in MATH 126. Offered: AWSpS.

Prerequisite: Math 126

Coordination: Department of Mathematics

Required: Yes

Course Topics:

Week	Topics	Resources
1:	Sec. 1.1: Lines and Linear Equations Sec. 1.2: Linear Systems and Matrices	Conceptual Problems for Chapter 1: PDF
2:	Sec. 2.1: Vectors Sec. 2.2: Span Sec. 2.3: Linear Independence	Conceptual Problems for Chapter 2: PDF
3:	Sec. 2.3: Linear Independence Sec 3.1: Linear Transformations	
4:	Sec 3.1: Linear Transformations MIDTERM 1	MIDTERM 1 ARCHIVE

5:	Sec. 3.2: Matrix Algebra Sec. 3.3: Inverses	Conceptual Problems for Chapter 3: PDF
6:	Sec. 4.1: Intro to Subspaces Sec. 4.2: Basis and Dimension	
7:	Sec. 4.3: Row and Column Spaces Sec 4.4: Change of basis Sec. 5.1: The Determinant Function	Conceptual Problems for Chapter 4: PDF
8:	Sec. 5.1: The Determinant Function MIDTERM 2 Sec. 5.2: Properties of the Determinant	MIDTERM 2 ARCHIVE Conceptual Problems for Chapter 5: PDF
9:	Sec. 6.1 Eigenvalues and Eigenvectors Sec. 6.2: Diagonalization	
10:	Sec. 6.2: Diagonalization (Optional) Sec. 6.3: Complex eigenvalues and eigenvectors Review	Conceptual Problems for Chapter 6: PDF
11:	FINAL EXAM	FINAL EXAMS

PHYS 121 Mechanics

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

Textbooks: - "Principles and Practice of Physics" by Erik Mazur

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. Offered: AWSpS.

Required: Yes

Coordination: Department of Physics

Course Topics

1 Mathematical reasoning
2 Acceleration in one dimension One-dimensional Kinematics
3 Systems and momentum Free-fall
4 Kinetic and internal energy Momentum and Collisions
5 Forces and Newton's Laws Relativity and Energy
6 Work and Conservation of Energy Introduction to Force
7 Potential energy diagrams Newton's Second Law
8 Motion in two-dimensions Rotational Kinematics
9 Dynamics of rigid bodies Torque Balance & Rotational Dynamics
10 Rolling and slipping

PHYS 122 Electromagnetism

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

Textbooks: - "Principles and Practice of Physics" by Erik Mazur

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:

Intro/Electrostatics
Coulomb Law
Electric Field
Field Lines and Symmetry
Gauss's Law
Electric Potential
Capacitance
Dielectrics
Circuits
Single-loop Circuits
Multiloop Circuits
Magnetic Fields
Current and Magnetism
Special Relativity
Unification of E & M, Ampere Law
Faraday Law
Induced emf
Inductance
Maxwell-Faraday Equation
EM Waves
AC Circuits

PHYS 123: Waves

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

Textbooks: - "Principles and Practice of Physics" by Erik Mazur

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:

- Periodic motion
- Energy and force in SHM
- SHM for springs and pendula
- 1D waves propagation
- Standing Waves
- Sound
- Interference & diffraction
- Intensity and Beats
- Doppler & Shocks
- EM Waves and Light rays
- Images, Mirrors & Lenses
- Ray Diagrams & Equations
- Practical Ray Optics
- Optics to Waves & Particles
- Thin Films & Multiple Slits
- Phasors, Waves & Particles
- Real Slits, Photons & Particles
- Static Fluids
- Fluid Dynamics
- Viscosity & Probability
- Probability and Entropy
- Entropy and Energy
- Thermal Processes & Heat

- PV Diagrams, Work & Energy
- Special Gas Processes
- Energy Conversion & Entropy
- Entropy & Gas Cycles

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

Credits: 4 Credits (4 hrs lecture, 1 hour quiz section)

Textbook: *Applied Statistics for Engineers and Scientists (3rd Edition)*, by Devore, Farnum, and Doi.

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. Cannot be taken for credit if credit received for STAT509/CS&SS 509/ECON 580. Prerequisite: either MATH 126 or MATH 136. Offered: AWSpS.

Required: No

Coordination: Department of Mathematics

Course Topics:

Week 1:

Lecture 1 (Ch. 1) 4/1; Descriptive vs. Inferential statistics, and types of data.

Lecture 2 (Ch. 1) 4/3; More on types of data, and histograms.

Lecture 3 (Ch. 1) 4/4; More on histograms.

Lecture 4 (Ch. 1) 4/5; Distributions.

Week 2:

Lecture 5 (Ch. 1) 4/8; More distributions; area under $N(0,1)$.

Lecture 6 (Ch. 1, 2) 4/10; Area under $N(\mu, \sigma)$, percentile/quantile, comparative boxplots.

Lecture 7 (Ch. 1) 4/11; Derivation of Binomial, and Poisson.

Lecture 8 (Ch. 2) 4/12; Sample mean, sample variance, sample standard deviation, and population/distribution mean.

Week 3:

Lecture 9 (Ch. 2) 4/15; Mean and variance of named distributions.

Lecture 10 (Ch. 2,3) 4/17; "Prob of x within some standard deviation of the mean" for ANY distr., qqplot, and scatterplot.

test1 4/19.

Week 4:

Lecture 11 (Ch. 3) 4/22; correlation coefficient, r .

Lecture 12 (Ch. 3) 4/24; Regression.

Lecture 13 (Ch. 3) 4/26; ANOVA in Regression.

Week 5:

Lecture 14 (Ch. 3) 4/29; regression on transformed data, polynomial regression, and overfitting.

Lecture 15 (Ch. 3) 5/1; multiple regression; interaction and collinearity.

Lecture 16 (Ch. 5) 5/2; Sampling distribution (of the sample mean).

Lecture 17 (Ch. 7) 5/3; CI for μ_x .

Week 6:

Lecture 18 (Ch. 7) 5/6; CI for μ_x , 1-sided CI for μ_x , and sample size estimation.

Lecture 19 (Ch. 7) 5/8; CI for π_x , and 2-sample CI for $\mu_1 - \mu_2$ and $\pi_1 - \pi_2$.

Lecture 20 (Ch. 7) 5/9; CIs for paired data, and CI's based on the t-distribution.

Lecture 21 (Ch. 8) 5/10; hypothesis testing with p-values.

Week 7:

Lecture 22 (Ch. 8) 5/13; More on hyp testing, and 2-sample tests.

Lecture 23 (Ch. 8) 5/15; Type I/II errors, meaning of α , and the chi-squared test of multiple proportions.

test2 5/17.

Week 8:

Lecture 24 (Ch. 8,9) 5/20; More on chi-squared test of k proportions, and the F test of k means.

Lecture 25 (Ch. 11) 5/22; Inference on the slope parameter in simple regression.

Lecture 26 (Ch. 11) 5/23; Inference in multiple regression: t-test and F-test of model utility.

Week 9:

Lecture 27 (Ch. 11) 5/28; CI, PI.

Lecture 28 (Ch. 5) 5/30; df, and set theory.

Lecture 29 (Ch. 5) 5/31; unconditional and conditional probs, and Bayes' theorem.

Week 10:

Lecture 30 (Ch. 5, 11) 6/3; Independence; neural networks.

Lecture 31 (Ch. 11) 6/5; Regression and Classification.

test3 6/7.

STAT 391: Quantitative introductory statistics for data science

Credits: 4 Credits (4 hrs lecture)

Textbook: *None.*

Catalog Description: The basic concepts of statistics, machine learning and data science, as well as their computational aspects. Statistical models, likelihood, maximum likelihood and Bayesian estimation, regression, classification, clustering, principal component analysis, model validation, statistical testing. Practical implementation and visualization in data analysis. Assumes knowledge of basic probability, mathematical maturity, and ability to program. Prerequisite: either CSE 312, or STAT 394/MATH 394 and STAT 395/MATH 395. Offered: Sp.

Required: No

Coordination: Department of Statistics

Learning Objectives/Course Topics:

- Statistical Learning
- Linear Regression
- Classification
- Resampling Methods
- Linear Model Selection and Regularization
- Support Vector Machine
- Unsupervised Learning
- Tree-Based Methods

Required

CSE 142 Computer Programming

Credits

4.0, (3 hrs lecture, 1 hr section)

Lead Instructor

Stuart Reges

Textbook

• *Building Java Programs: A Back to Basics Approach*, Reges & Stepp

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

- (1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
- (2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, , and economic factors (H)
- (7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies (H)

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: A Back to Basics Approach, Reges & Stepp

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

- (1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
- (2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, , and economic factors (H)
- (7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies (H)

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:

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

ABET Outcomes

(1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)

(6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions (M)

(7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies (H)

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. Examines fundamentals of enumeration and discrete probability; applications of randomness to computing; polynomial-time versus NP; and NP-completeness. Prerequisite: CSE 311.. Prerequisite: CSE 311.

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

(1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics(H)

(6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions (M)

(7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies (H)

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 Structures

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; multithreading and parallel algorithms; P and NP complexity classes. No credit if CSE 373 has been taken

Prerequisites

CSE 311

CE Major Status

Required

Course Objectives

communicate the representation and organization of data in terms of ubiquitous computing abstractions such as stacks, queues, trees, hash-tables, and graphs

analyze algorithms for correctness and efficiency, including the use of asymptotic analysis

design parallel programs that use extra computational resources to complete a task more quickly

recognize software errors related to concurrent execution of tasks such as race conditions

create software that implements classic data structures and algorithms and uses such algorithms appropriately

ABET Outcomes

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)

2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (H)

6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions (H)

7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (H)

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 (2 lecture hours)

P, NP, NP-Complete (1 lecture hour)

CSE 351 The Hardware/Software Interface

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Luis Ceze

Textbook

Computer Systems, Bryant & O'Hallaron

C Programming Language, Kernighan & Ritchie

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:

- understand the multi-step process by which a high-level program becomes a stream of instructions executed by a processor;
- know what a pointer is and how to use it in manipulating complex data structures;
- 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);
- understand the basic organization and parameters of memory hierarchy and its importance for system performance;
- be able to explain the role of an operating system;
- know how Java fundamentally differs from C;
- grasp what parallelism is and why it is important at the system level; and
- be more effective programmers (more efficient at finding bugs, improved intuition about system performance).

ABET Outcomes

- (6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions
- (7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies

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 369 Intro to Digital Design

Credits

2.0 (1.5 hrs lecture)

Lead Instructor

Justin Hsia

Textbook

The **optional** text is the *third edition* of Brown and Vranesic's *Fundamentals of Digital Logic with Verilog Design* book ("B&V"), ISBN **9870073380544**. **Course Description**

Course Description:

Introduces the implementation, specification, and simulation of digital logic. Boolean algebra; combinational circuits including arithmetic circuits and regular structures; sequential circuits including finite-state-machines; and use of field-programmable gate arrays (FPGAs).

Emphasizes simulation, high-level specification, and automatic synthesis techniques. No credit if EE 271 has been taken.

Prerequisites

CSE 143, 311

CE Major Status

Required

Course Objectives

1. *Design and implement* digital circuits and systems in the laboratory using fundamental concepts.
2. *Write* Boolean equations for basic combinational logic circuits, use Boolean algebra to simplify such equations, then implement the resulting designs in the laboratory.
3. *Design and implement* combinational circuits of medium complexity in the laboratory.
4. *Design and implement* basic sequential circuitry and finite state machines in the laboratory.
5. *Identify* real world timing problems in both combinational and sequential circuits and design basic digital systems that are tolerant of such effects.
6. *Design and implement* combinational and sequential circuits using programmable logic devices.
7. *Develop* basic structural models of digital systems using the Verilog hardware design language.

ABET Outcomes

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (H)
3. an ability to communicate effectively with a range of audiences (M)
4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts (M)
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (H)
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions (H)
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (H)

Course Topics

- Combinational Logic
- Verilog Basics
- Karnaugh Maps
- Sequential logic
- Finite state machines (FSMs)
- FSM Design, Multiplexors, Adders
- Encoders, Decoders, Registers, Counters
- Project Tips
- Computer Components, FPGAs

EE 205 Introduction to Signal Conditioning

Title: Introduction to Signal Conditioning

Credits: 4

UW Course Catalog Description

Coordinator: Alexander Mamishev, Professor, Electrical and Computer Engineering

Goals: 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 basic elements 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.

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

- *Describe* the role of signal conditioning in digital systems.
- *Design* voltage divider circuits to attenuate voltage signals for digital input.
- *Design* simple op amp circuits to amplify sensor outputs, including saturation and slew rate considerations.
- *Describe* techniques to deal with noisy sensors.
- *Design* simple high pass and low pass passive and active filters.
- *Acquire* and *analyze* analog signals in Matlab.
- *Analyze* simple circuits using PSpice.
- *Determine* the sample rates necessary for signals with specific frequency content and *describe* the effects of improper sample rates.
- *Design* anti-aliasing filters.
- *Explain* analog transmission line effects on digital signals.
- *Design* terminations for digital signal lines.
- *Describe* the importance of isolation.
- *Design* isolation methods.
- *Sketch* a simple control system block diagram and *explain* its basic operation.

Textbook: Horowitz and Hill, ***The Art of Electronics***, 3rd Ed., Cambridge University Press, 2015, ISBN-13: 978-0521809269.

Reference Texts:

- Pallas-Areny and Webster, ***Sensors and Signal Conditioning***, 2nd Ed., John Wiley & Sons, 2001.
- Hayes and Horowitz, ***Learning the Art of Electronics: A Hands-On Lab Course***, 2016, ISBN-13:978-0521177238.
- Nilsson and Riedel, ***Electric Circuits***, 11th Ed., Prentice Hall, 2011.

Prerequisites by Topic:

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

Topics:

- Sensors, signals, A/D conversion, signal conditioning paradigm, voltage, current.
- Attenuation, Ohm's Law, resistance, voltage divider, resistor precision, power dissipation.
- Amplification, ideal op amps, inverting amplifier, non-inverting amplifier, saturation, slew rate, bandwidth, impedance matching.
- Strain gauges, noise, frequency domain, filtering, FFT, low pass passive filter design, PSpice.
- High pass passive filters, phase angle, active filters.
- Aliasing, Nyquist limit, anti-aliasing filters, sample and hold.
- Digital signals and transmission lines; attenuation, ringing, termination, cross talk.

ABET Student Outcome Coverage: This course addresses the following outcomes:

H = high relevance, M = medium relevance, L = low relevance to course.

(1) *An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)* The homework, exams and laboratories require direct application of mathematical, scientific, and engineering knowledge to solve circuit theory problems and analyze, design and test signal conditioning circuits.

(3) *An ability to communicate effectively with a range of audiences (M)* Students are required to write and submit a formal laboratory report for each experiment describing the circuits used and the results achieved.

(4) *An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts (M)* Several examples in the lectures are based on well known books and movies, based on which different engineering methods are solutions are considered. An ensuing discussion includes comparison of pluses and minuses of different approaches.

(5) *An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (M)* Laboratory work is carried out in teams of typically three students.

(6) *An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions (M)* Students conduct simple signal conditioning experiments using personal multimeters, a breadboard and a parts kit, and a PC-based oscilloscope. The experiments require student to account for differences between measured data and predictions. Some freedom of experiment design is provided.

(7) *An ability to acquire and apply new knowledge as needed, using appropriate learning strategies (M)* Course materials draw on a mix of books, slides, web entries, live demonstrations, software demonstrations, and recorded lectures. Learning approaches include group and individual discussions, review of materials, verification of validity of web-based information, and estimation of the realism of the solutions.

Prepared By: Alex Mamishev (earlier version by Rich Christie)

Last Revised: March 18, 2019

EE 371: Design of Digital Circuits and Systems

Title: Design of Digital Circuits and Systems

Credits: 5 (4 lecture; 1 lab)

UW Course Catalog Description

Coordinator: James K. Peckol, Principal Lecturer, Electrical and Computer Engineering

Goals: The execution of modern digital electronic systems designs presents challenges that demand new ways of thinking about such problems. Building upon the fundamental concepts of electronic circuits and those developed in EE 271, the main objective of EE 371 is to provide students with a theoretical background to and practical experience with the tools, techniques, and methods for solving challenges related to modeling complex systems using the Verilog hardware description language (HDL), signal integrity, managing power consumption in digital systems, and ensuring robust intra and inter system communication.

We will work with the Altera DE1-SOC development board that utilizes the Cyclone V FPGA combined with a variety of peripheral devices, including the embedded NIOS II processor, as our target hardware platform. The hardware side of the applications will be specified then designed, modeled, and tested using the Verilog HDL and the libraries and tools provided under the Quartus II development environment. We will synthesize then download the tested modules onto the DE1-SOC board where they will be integrated into a complete working system. The software side of the applications will be written in C, cross-compiled under the NIOS II IDE then downloaded and executed on the embedded NIOS II processor.

Upon completion of the class the student will have developed strong design skills for implementing complex digital logic systems in modern design languages onto FPGAs and similar programmable fabrics.

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

- *Identify and understand* real-world timing problems in both combinational and sequential circuits,
- *Understand and recognize* the parasitic elements of circuit traces, the interactions between traces, and the affects of such interactions as well as affects from the outside world on circuit signal integrity.
- *Understand and design* intra and inter system communication and timing in systems comprising components operating with differing (asynchronous) timing, multiple clocks, and clocking schemes.
- *Design, model, and implement* intermediate level digital systems that are tolerant of real-world timing effects.
- *Design and implement* schemes to measure, manage, and reduce power consumption in digital designs.
- *Understand and design* systems comprising networks of distributed components or subsystems.
- *Design and Implement* a system supporting a contemporary network interface standard.

Textbook: John F. Wakerly, *Digital Design Principles and Practices*, Prentice-Hall, 4th Ed., 2006.

Prerequisites:

- either EE-205 or EE-215
- either EE-271 or CSE-369
- **Topics:**
 - Verilog HDL and HDL Modeling at the Dataflow and Behavioral levels, 2.0 weeks
 - Datapath and Control Components in a digital system, 1.0 week
 - Timing and Practical Considerations, 1.0 week
 - Introduction signal integrity and signal integrity issues, 2.0 weeks
 - Power management in digital systems, 1.0 week
 - Introduction to networks and basic network concepts, 2.0 weeks
 - Testing and design for test in digital systems, 1.0 week

Course Structure: The course meets for 4 hours of lecture and 3 hours of laboratory.

Computer Resources: There will be extensive computer usage in the homework and laboratories for design and simulation utilizing the Verilog HDL and FPGA device software development packages.

Laboratory: There are weekly laboratory projects. For each such project, the students have to design the circuit, construct it and demonstrate it to the instructor and/or teaching assistant. In all of the projects, the students use programmable logic devices and microprocessors for implementation with the designs developed using the Verilog HDL. All laboratories are done in an open lab as two or three person teams.

Grading: The grade is based upon weekly homework assignments, the laboratory projects, midterm exams, and a comprehensive final examination.

ABET Student Outcome Coverage: This course addresses the following outcomes:

H = high relevance, M = medium relevance, L = low relevance to course.

(1) *An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.* **(H)** These are all done as an integral and routine part of the material taught. Theory is presented a standard part of the homeworks, exams, and laboratories in the context of its application to real-world problems and its limitations under real-world constraints.

(2) *An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.* **(H)** Each of the laboratory projects specifies and assigns a particular design problem to be solved.

(6) *An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.* **(H)** Students will use modern computers, test instrumentation, modeling, and simulation tools. A significant component of designing and developing a real-world application is ensuring that one's system performs to specification in the intended environment. Such assurance can only be gained by testing the system in such a context then analyzing the results of those tests. Such a process is integral to this class, to each of the labs and to the final project.

Prepared By: James K. Peckol, Gaetano Borriello, and Scott Hauck

Last Revised: 05/02/19

CE System Electives

CSE 401 Introduction to Compiler Construction

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Hal Perkins

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

CSE 332; 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

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (H)
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions (H)
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (H)

Course Topics

- Regexp and scanners
- Grammars and ambiguity (3.1-3.3)

- LR (bottom-up) parsing
- Parser Construction
- LR table construction
- LR conflicts, first
- ASTs & visitors;
- LL Parsing & recursive descent (3.3)
- Intro to semantics and type checking (4.1-4.2)
- Semantics; Attribute grammars (4.3)
- Symbol tables and representation of types
- x86-64 (everything you forgot from 351)
- Code shape - objects and dynamic dispatch
- Finish codeshape
- IRs

CSE 402 Design and Implementation of Domain-Specific Languages

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

Ras Bodik

Textbook

JavaScript, the good parts

JavaScript, the Definitive Guide

- JS is our implementation language

Mastering Regular Expressions

- more than you need to know about this topic

The Definitive ANTLR 4 Reference

- ANTLR is our parser

Programming in Lua

- how to implement objects; how to use coroutines

Course Description

Design and implementation of domain-specific languages. Creation of new programming abstractions, formal and informal language specification techniques, implementation strategies to support language analysis and execution on traditional and non-traditional computing platforms. Selection and use of appropriate software tools and development environments to build novel DSLs.

Prerequisites

CSE 332; CSE 351

CE Major Status

Selected Elective

Course Objectives

Programming languages bridge two levels of abstraction: (i) the world of the programmer who uses concepts of his domain, such as graphs or matrices, and (ii) the world of the computer, which prefers abstractions that can be efficiently executed, such as arrays, threads, and remote procedure calls.

This course will teach you how languages translate the programmer abstraction into the executable abstractions. We will talk about compilers, interpreters, parsers, and optimizations. The lectures will cover both general-purpose languages and the recent modern languages

designed for a specific domain task. The programming assignments will focus on the domain-specific languages.

ABET Outcomes

- (1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
- (3) an ability to communicate effectively with a range of audiences
- (5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (H)
- (7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (H)

Course Topics

- Part I: Implementation Strategies / Regular Expressions
- Part II: Optimizations / Query Languages
- Part III: Parsing / Templating
- Part IV: Reactive Languages, Laziness / Working with Data
- Part V: Types and Objects, Garbage Collection, Staging

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

CSE 331; CSE 332

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

- (1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
- (2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, , and economic factors
- (3) an ability to communicate effectively with a range of audiences

- (4) an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
- (5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives
- (6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions
- (7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies

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 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

CSE 312; CSE 331; 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

- (1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
- (2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, , and economic factors
- (3) an ability to communicate effectively with a range of audiences
- (4) an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts

- (5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives
- (6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions
- (7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies

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

Credits

5.0 (5 hrs lecture/meeting times)

Lead Instructor

Katherina Reinecke

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

- (1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
- (2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, , and economic factors
- (3) an ability to communicate effectively with a range of audiences
- (4) an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
- (5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives
- (6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions
- (7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies

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

Internals of a relational database management systems (DBMS). Architecture of a DBMS. Data storage and indexing. Query execution, buffer management, operator algorithms. Query optimization. Transactions: recovery and concurrency control. Parallel query evaluation. Distributed transactions. Data replication. NoSQL and NewSQL systems. Students build a medium size DBMS in Java.

Prerequisites

CSE 332; CSE 344.

CE Major Status

Selected Elective

Course Objectives

This course teaches the internals of relational database management systems and how to build such systems. The course covers the full stack. Starting from a quick review of the relational model and SQL, the course dives into the architecture of a DBMS and then goes in depth into each component: storage manager, query executor, query optimizer, transactions, parallel processing, distributed transactions, data replication, architecture of NoSQL and NewSQL systems, etc. Students get to implement each of the components in their own Java DBMS. At the end of the quarter, successful students have a working DBMS that can execute SQL queries, update data, run concurrent transactions, recover from failures, and either execute in parallel or optimize queries.

ABET Outcomes

- (1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
- (3) an ability to communicate effectively with a range of audiences

(6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions

(7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies

Course Topics

- data models
- conceptual design
- query languages
- system components
- data storage
- query optimization
- transaction processing.
- parallel, shared-nothing processing
- distributed transactions
- data replication

CSE 451 Introduction to Operating Systems

Credits

4.0 (3 hrs lecture, 1 hr section)

Lead Instructor

John Zahorjan

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; 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

- (1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
- (2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, , and economic factors
- (3) an ability to communicate effectively with a range of audiences
- (4) an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
- (5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

(6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions

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

Arvind Krishnamurthy

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

CSE 332 and 333, 451 recommended

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

- (1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
- (2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, , and economic factors
- (5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives
- (6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions

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 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

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (H)
3. an ability to communicate effectively with a range of audiences (H)

4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts (H)
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (H)
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions (H)
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (H)

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 Networking: A Top-Down Approach (6th Edition)*, Kurose and Ross

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

This course introduces the basics of networking, ranging from sending bits over wires to the Web and distributed computing. We focus on the networking ground between these two extremes, particularly focusing on the engineering of the Internet - goals, constraints, solutions, and experiences. The outcome of this course for you should be an appreciation of the fundamental challenges of networking, design strategies of proven value, and common implementation technologies.

ABET Outcomes

- (1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
- (2) an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, , and economic factors
- (5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

(6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions

Course Topics

Topics will include: framing, error correction, packet switching, multi-access (Ethernet), addressing and forwarding (IP), distance vector and link state routing, queueing and scheduling, reliable transport, congestion control (TCP), quality of service, naming (DNS), and security.

CSE 469 Computer Architecture I

Credits

5.0 (3 hrs lecture, 2 hours meetings)

Lead Instructor

Scott Hauck

Textbook

Patterson, Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, ARM Edition, Morgan Kaufmann, 2017.

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

CSE 143, CSE 369.

CE Major Status

Selected Elective

Course Objectives

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

Write simple assembly language programs.

Identify the major components of a microprocessor.

Design a microprocessor that supports a given instruction set.

Analyze microprocessor performance, including cache memory systems.

Demonstrate understanding of modern microprocessor features.

ABET Student Outcome Coverage: This course addresses the following outcomes:

H = high relevance, M = medium relevance, L = low relevance to course.

(1) *An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)*

(2) *An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (M)*

(6) *An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions (M)*

(7) *An ability to acquire and apply new knowledge as needed, using appropriate learning strategies (L)*

Topics:

Introduction to processor architecture. Performance measures.

Assembly language programming.

Computer arithmetic.

Processor datapaths, control.

Pipelining.

Memory hierarchy, caches.

Advanced topics in computer architecture: ILP, VLIW, Superscalar

CSE 470 / EE 470 Computer Architecture II

Credits

5.0 (3 hrs lecture, 2 hours meetings)

Lead Instructor

Luis Ceze

Textbook

Hennessy, Patterson. *Computer Architecture: A Quantitative Approach* (5th Edition, Morgan Kaufmann, 2012)

Course Description

Advanced computer architecture. Performance evaluation and energy efficiency. Instruction set architectures. Instruction-level parallelism. Modern microprocessor micro-architecture. Thread-level parallelism. Cache coherency and memory consistency in shared-memory multiprocessors. Memory hierarchy. GPU architecture. Warehouse-scale computing. Trends in computer design. Prerequisite: CSE351; either CSE 469 or EE 469.

Prerequisites

CSE 351; either CSE 469 or EE 469

CE Major Status

Selected Elective

Course Objectives

Understand how a modern microprocessor works.
Understand how multicores work.
Have a general understanding of GPUs.
Have a general understanding of Warehouse-scale computers.
Appreciate the importance of energy efficiency in computing.

ABET Student Outcome Coverage: This course addresses the following outcomes:

H = high relevance, M = medium relevance, L = low relevance to course.

(1) *An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)*

(2) *An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (M)*

(6) *An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions (M)*

(7) *An ability to acquire and apply new knowledge as needed, using appropriate learning strategies (L)*

Topics:

Introduction to architecture and metrics (performance and energy)

The ISA

Pipelining

Branch Prediction

Superscalars/Dynamic Scheduling

Multithreading

Memory Hierarchy (caches, prefetching, virtual memory)

Cache Coherence

Memory Consistency

Overview of GPUs and Warehouse-scale computers

Current trends in computer architectures (e.g., specialization).

CSE 474 Introduction to Embedded Systems

Credits

4 (4 hours lecture, 1 hour recitation)

Lead Instructor

Shwetak Patel

Textbook

Embedded Systems - A Contemporary Design Tool - James K. Peckol

Course Description

Introduces the specification, design, development, and test of real time embedded system software. Use of a modern embedded microcomputer or microcontroller as a target environment for a series of laboratory projects and a comprehensive final project.

Prerequisites

CSE 143

CE Major Status

Selected Elective

Course Objectives

- *Understand* the embedded systems development cycle and *use* the tools supporting such development.
- *Understand* the importance of and methods for designing and developing safe, reliable, and fault tolerant embedded systems.
- *Develop* high quality, well annotated, multfile software packages.
- *Understand* and *design* systems subject to real-time constraints.
- *Develop* drivers for basic analog and digital peripheral devices.
- *Use* debugging tools to verify proper program execution on a dedicated target system.
- *Test* a program executing on a dedicated target system.

ABET Student Outcome Coverage: This course addresses the following outcomes:

H = high relevance, M = medium relevance, L = low relevance to course.

(1) *An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics*

(2) *An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (M)*

(4). *an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts (H)*

(6) An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions (M)

Topics:

Software design and abstraction

Software development life cycle

C programming for embedded systems

Control and control flow

Tasks, control, and scheduling

Inter-task communications

Operating systems basics

Memory management

Basic digital signal processing techniques: digital filters, FFTs

System performance and optimization

Safety, reliability, fault tolerance, and test

System testing

Advanced topics in embedded systems

Note: recitation will be used for a TA-led introduction and support on writing C programs.

Recitation is optional, but strongly recommended for students without prior C programming experience.

CSE 475 Embedded Systems Capstone

Credits

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

Lead Instructor

James Peckol

Textbook

John F. Wakerly, *Digital Design Principles and Practices*, Prentice-Hall, 4th. ed., 2006.

Course Description

Capstone design experience. Prototype a substantial project mixing hardware, software, and communications. Focuses on embedded processors, programmable logic devices, and emerging platforms for the development of digital systems. Provides a comprehensive experience in specification, design, and management of contemporary embedded systems.

Prerequisites

either E E 271 or CSE 369; either CSE 466, E E 472, or CSE 474/E E 474. Offered: jointly with E E 475

CE Major Status

Selected Elective

Course Objectives

- *Introduce and explain* in-depth embedded systems concepts, design principles, practices, and techniques, and then *apply and practice* such methods in real-world applications.
- *Introduce* the system design and development process and steps.
- *Practice* written and oral communications skills.
- *Utilize* concepts studied in the design and development of several real-world contemporary digital electronics, software, and computer systems projects then ultimately bringing together in the design and development of a self-proposed comprehensive capstone design project.

ABET Outcomes

(1) *An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)* These are all done as an integral and routine part of the material taught. Theory is presented as a standard part of the homeworks, exams, and laboratories in the context of its application to real-world problems and its limitations under real-world constraints. The laboratory exercises require the student to assess and analyze the assignment, then apply basic engineering knowledge to either solve the problem or state why (based upon their analysis) they are unable to fully satisfy the requirements. For each of the lab projects, the student must analyze the requirements, then design, implement, and test a hardware/software system that meets the stated requirements. The student must then propose a test plan and demonstrate that their design meets the initial requirements. The final project requires the application of such knowledge to a project of the student's own choice.

(2) *An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (H)* Students design and implement two significant real world systems followed by a substantial final project. Each of the two laboratory projects provides a high-level requirements specification for the problem to be solved. For the final project, the students must develop their own requirements and design specifications.

(3) *An ability to communicate effectively with a range of audiences (M)* Teams must research, prepare a formal report on a topic of current interest, and present their report to the class. They must also prepare an oral presentation to the class describing their final project, discussing any problems and how they were solved, and proposing how they might alter their design should they begin again.

(4) *An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts. (H)* Ethics and professional behavior are strongly stressed throughout the course. Considered areas include copyrights, national and international patents, licensed material, intellectual property, plagiarism, citing sources for material or idea, and using published algorithms and designs. Projects, lab assignments, and research projects that do not cite sources are given failing marks. Lecture material routinely stresses the need for designs to consider international markets and the need to satisfy international standards, including those for safety and health.

(5) *An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives. (M)* Although not multidisciplinary since the class is in the student's selected major, the students work as members of 2-3 person teams to execute each of the labs and the final project.

(6) *An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions. (H)* Students will use modern computers, development tools and debugging techniques. A significant component of designing and

developing a real world application is ensuring that one's system performs to specification in the intended environment. Such assurance can only be gained by testing the system in such a context then analyzing the results of those tests. Such a process is integral to this class, to each of the labs and to the final project.

(7) *An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.* (M) Discussions of contemporary technologies, corporate needs and responsibilities, the legal impacts of designs, and the ever-evolving engineering discipline are an integral part of the lecture material. Lecture material continually emphasizes that today's technology is transitory and that the students must learn the basics so that these may form a foundation upon which they will build future technologies.

Course Topics

- Introduction to Basic Laboratory Tools and Techniques
- System Specification, Modeling, and Design
- Review of Analog and Digital Concepts
- Programmable Logic Devices
- Busses and Networks
- Signal Integrity
- High Speed Signals and Signal Management
- Memory Systems
- Reliability, Fault Tolerance, and Test
- Topics of Current Interest

CSE 481A Capstone Software Design: Operating Systems

Credits

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

Lead Instructor

Xi Wang

Textbook

- Hardware and Software Support for Virtualization, by Edouard Bugnion, Jason Nieh, and Dan Tsafir

Course Description

Students work in substantial teams to design, implement, and release a software project involving multiple areas of the CSE curriculum. Emphasis is placed on the development process itself, rather than on the product. Teams are expected to develop a work plan, and to track and document their progress against it.

Prerequisites

CSE 332; CSE 351; either CSE 331 or 371.

ABET Outcomes

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (H)
3. an ability to communicate effectively with a range of audiences (H)

4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts (H)
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (H)
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions (H)
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (H)

Course Topics

[CSE 481A](#) is a capstone course on operating systems, with an emphasis on virtualization. In lectures we will introduce *lvisor*, a minimal x86 hypervisor based on KVM. We expect you to work on a project related to *lvisor*, finish exercises, discuss research papers, and make a presentation of your project at the end of the quarter

CSE 481C Capstone Software Design: Robotics

Credits

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

Lead Instructor

Maya Cakmak

Textbook

None

Course Description

Student teams design and implement a software project involving multiple areas of the CSE curriculum. Course emphasizes the development process, rather than the product.

Prerequisites

CSE 332; CSE 351; either CSE 331 or 371.

CE Major Status

Selected Elective

Course Objectives

The main goal of this course is to open up new career options in robotics for computer science and engineering students. To that end, the course will teach you the basics of robotics and give you implementation experience. You will learn to use libraries and tools within the most popular robot programming framework ROS (Robot Operating System). We will touch on robot motion, navigation, perception, planning, and interaction through mini-lectures, labs and assignments, eventually integrating these components to create autonomous or semi-autonomous robotic functionalities. The project will give you team-work experience with large scale software integration and it will get you thinking about opportunities for using robots to make people's lives easier. At the end of the quarter students are expected to:

- Understand basics of robot navigation, perception, planning, interaction; have a sense of challenging problems in robotics
- Know how to use important tools in ROS, be able to contribute to ROS, have awareness of available packages in ROS

- Be comfortable operating a robot platform, have experience using ROS tools to control a robot platform
- Understand the importance of interface design and robustness of functionalities in robotics
- Be prepared to interview for a robotics job

ABET Outcomes

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (H)
3. an ability to communicate effectively with a range of audiences (H)
4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts (H)
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (H)
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions (H)
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (H)

Course Topics

- Need finding for technology-based solutions
- Building a lean business model for technology startups
- Sketching and storyboarding
- Robot Operating System
- Large scale software integration
- Robot navigation and localization
- Robot manipulation
- Robot perception
- CAD and fabrication basics

CSE 481N NLP

Credits

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

Lead Instructor

Yejin Lee

Textbook

None

Course Description

Student teams design and implement a software project involving multiple areas of the CSE curriculum. Course emphasizes the development process, rather than the product.

Prerequisites

CSE 332; CSE 351; either CSE 331 or CSE 352.

CE Major Status

Selected Elective

Course Objectives

This class will provide students with an intensive 10-week experience in successfully completing a challenging, but well-scoped research project.

Participants will work in small groups (2-3 people in each group) to hone their technical skills to quickly absorb and adapt new technical knowledge, gain experience in complex programming, perform thorough experiments and analysis, and learn how to find a path when faced with negative results.

Additional objectives of this class include:

- (1) technical communication skills to produce high quality interim technical reports that inspire insightful discussion across project groups,
- (2) advisory project experience to provide technical advice and constructive feedback on others, and

(3) project management skills to prioritize work items to maximize the chance for successful outcome.

ABET Outcomes

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (H)
3. an ability to communicate effectively with a range of audiences (H)
4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts (H)
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (H)
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions (H)
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (H)

Course Topics

Areas of applications include text classification, information extraction, social media analysis, summarization, conversation (usually called dialogue by researchers), interpretation of deep neural models, question answering, and semantic parsing. In addition, cross-disciplinary applications are also encouraged, for example, image captioning, code generation from natural language descriptions, language based robot manipulation and navigation, connecting NLP with various disciplines such as computer vision, robotics, HCI, and programming languages.

CSE 481S Security

Credits

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

Lead Instructor

Yoshi Kohno and Franz Roesner

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 484

CE Major Status

Selected Elective

Course Objectives

Student teams will be tasked with creating a computer security themed product. The work will progress from product conception to requirements to design to implementation to evaluation. Along the way, students will incorporate key computer security tools and practices, including threat modeling, penetration testing, and bug fixing. Examples include password managers, censorship resistance systems, and mobile payment systems.

ABET Outcomes

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (H)

3. an ability to communicate effectively with a range of audiences (H)
4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts (H)
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (H)
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions (H)
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (H)

Course Topics

- Ethics
- Threat modeling and risk management
- Software security
- Cryptography
- Web security
- HCI-security
- Network security
- Mobile device security
- Malware
- Advanced topics

CSE 481V AR/VR

Credits

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

Lead Instructor

Aditya Sankar

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 332; CSE 351; either CSE 331 or CSE 352.

CE Major Status

Selected Elective

Course Objectives

Virtual and Augmented reality are promising technologies that are certain to make an impact on the future of business and entertainment. In this capstone, students will work in small project teams to build applications and prototype systems using state of the art Virtual Reality (VR) and Augmented Reality (AR) technology. Seattle is a nexus of VR tech, with Oculus Research, Valve, Microsoft (hololens), Google (cardboard, jump), and teams in the area. We will be developing on the latest VR/AR headsets and platforms, and will bring in leading VR experts for lectures and to supervise student projects. Students will experience the end-to-end product cycle from design to deployment, and learn about VR/AR technology and applications. The capstone culminates in a highly anticipated demo day where the students demonstrate their creations to other students, faculty and industry luminaries.

ABET Outcomes

1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H)
2. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (H)
3. an ability to communicate effectively with a range of audiences (H)
4. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts (H)
5. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (H)
6. an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgement to draw conclusions (H)
7. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies. (H)

Course Topics

- Software development in teams
- 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.

Appendix B – Faculty Vitae

Anderson, Richard

1. Education – PhD, Computer Science, Stanford University, 1986.
2. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

1985-1986 Postdoctoral Research Fellow, Mathematical Sciences Research Institute, Berkeley, CA
1986-1991 Assistant Professor, 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
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
3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

2001-2002 Visiting Researcher, Learning Sciences and Technology Group, Microsoft Research
2009-2018 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 Subcommittee 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 (within and outside of the institution)

ICTD 2017 PC Chair
COMPASS 2018 PC Chair
COMPASS 2019 General Chair
8. Selected recent publications
 1. Samia Ibtasam, Lubna Razaq, Haider W. Anwar, Hamid Mehmood, Kushal Shah, Jennifer Webster, Neha Kumar, and Richard Anderson. 2018. Knowledge, Access, and DecisionMaking: Womens Financial Inclusion In Pakistan. In Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS '18).

2. Galen Weld, Trevor Perrier, Jenny Aker, Joshua E. Blumenstock, Brian Dillon, Adalbertus Kamanzi, Editha Kokushubira, Jennifer Webster, and Richard J. Anderson. 2018. eKichabi: Information Access through Basic Mobile Phones in Rural Tanzania. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18).
3. Sam Castle, Fahad Pervaiz, Galen Weld, Franziska Roesner, and Richard Anderson. 2016. Let's Talk Money: Evaluating the Security Challenges of Mobile Money in the Developing World. In Proceedings of the 7th Annual Symposium on Computing for Development (ACM DEV '16).
4. Trevor Perrier, Brian DeRenzi, and Richard Anderson. 2015. USSD: The Third Universal App. In Proceedings of the 2015 Annual Symposium on Computing for Development (DEV '15).
5. Trevor Perrier, Nicola Dell, Brian DeRenzi, Richard Anderson, John Kinuthia, Jennifer Unger, and Grace John-Stewart. 2015. Engaging Pregnant Women in Kenya with a Hybrid Computer-Human SMS Communication System. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15).
6. Samia Ibtasam, Hamid Mehmood, Lubna Razaq, Jennifer Webster, and Richard Anderson. 2017. An Exploration of Smartphone Based Mobile Money Applications in Pakistan. In Proceedings of the Ninth International Conference on Information and Communication Technologies and Development (ICTD 17).
7. Neha Kumar, Trevor Perrier, Michelle Desmond, Kiersten Israel-Ballard, Vikrant Kumar, Sudip Mahapatra, Anil Mishra, Shreya Agarwal, Rikin Gandhi, Pallavi Lal, and Richard Anderson. 2015. Projecting health: community-led video education for maternal health. In Proceedings of the Seventh International Conference on Information and Communication Technologies and Development (ICTD '15).
8. Neha Kumar, Waylon Brunette, Nicola Dell, Trevor Perrier, Beth Kolko, Gaetano Borriello, and Richard Anderson. Understanding sociotechnical implications of mobile health deployments in India, Kenya, and Zimbabwe, *Information Technologies & International Development*, 11(4), 2015.
9. Neha Kumar and Richard J. Anderson. 2015. Mobile Phones for Maternal Health in Rural India. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)
10. Briefly list the most recent professional development activities. None

Anderson, Ruth E.

1. 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:
 - Senior Lecturer, University of Washington, Paul G. Allen School of Computer Science & Engineering, Seattle, WA (Autumn 2017-present)
 - Full-Time Lecturer, University of Washington, Paul G. Allen School of Computer Science & Engineering, Seattle, WA (Autumn 2013-2017)
 - Part-Time Lecturer, University of Washington, Paul G. Allen School of Computer Science & Engineering, Seattle, WA (2006-2013)
 - 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 Teaching Award, awarded to one faculty member per year by students, UW, (2016)
 - ACM Faculty Award, awarded to one faculty member per year 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)
 - Inside Institution:
 - UW College of Engineering Council on Educational Policy, Autumn 2016-present
 - UW CSE Undergraduate curriculum committee, chair, Autumn 2018-present
 - UW CSE General TA Training, Winter 2017-present
 - UW CSE ACM-W chapter faculty advisor, 2014-present
 - UW CSE Undergraduate admissions committee, 2014-present
 - UW CSE PMP admissions committee, 2013-2014
 - Outside Institution:
 - Associate Program Chair, ACM Special Interest Group on Computer Science Education Symposium 2013-2019 (SIGCSE 2013-2015, 2017-2019)
 - Posters Co-Chair, ACM Special Interest Group on Computer Science Education Symposium 2020 (SIGCSE 2020)
 - Publications Chair, ACM Special Interest Group on Computer Science Education Symposium 2017, 2018 & 2019 (SIGCSE 2017, 2018 & 2019)
 - Local Arrangements Chair, ACM Special Interest Group on Computer Science Education Symposium 2017 (SIGCSE 2017)

- Panels & Special Sessions Chair, ACM Special Interest Group on Computer Science Education Symposium 2016 (SIGCSE 2016)
 - 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 (last 5 years)
- Austin Cory Bart, Kalpathi Subramanian, Ruth E. Anderson, and Nadeem Abdul Hamid. 2018. Preparing, Visualizing, and Using Real-world Data in Introductory Courses. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). ACM, New York, NY, USA, 676-677. DOI: <https://doi.org/10.1145/3159450.3159616>
 - Colleen M. Lewis, Ruth E. Anderson, and Ken Yasuhara. 2016. "I Don't Code All Day": Fitting in Computer Science When the Stereotypes Don't Fit. In Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16). ACM, New York, NY, USA, 23-32. DOI: <https://doi.org/10.1145/2960310.2960332>
 - Ruth E. Anderson, Michael D. Ernst, Robert Ordóñez, Paul Pham, and Ben Tribelhorn. 2015. A Data Programming CS1 Course. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15). ACM, New York, NY, USA, 150-155. DOI: <https://doi.org/10.1145/2676723.2677309>
 - Ruth E. Anderson, Michael D. Ernst, Robert Ordóñez, Paul Pham, and Steven A. Wolfman. 2014. Introductory programming meets the real world: using real problems and data in CS1. In Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14). ACM, New York, NY, USA, 465-466. DOI: <https://doi.org/10.1145/2538862.2538994>
 - Andrea Danyluk, Steve Roach, Elizabeth K. Hawthorne, Henry M. Walker, Ruth E. Anderson, and Christa M. Chewar. 2013. ACM/IEEE computer science 2013 exemplar-fest. In Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13). ACM, New York, NY, USA, 285-286. DOI: <https://doi.org/10.1145/2445196.2445284>
9. Briefly list the most recent professional development activities
- Consortium for the Advancement of Undergraduate STEM Education (CAUSE) Cohort Member, Autumn 2018-present

Anderson, Thomas E.

1. Education – degree, discipline, institution, year

A.B., Philosophy, Harvard University, 1983
M.S., Computer Science, University of Washington, 1989
PhD, Computer Science, University of Washington, 1991

2. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

ETH Zurich, Visiting Professor, 2019, full-time
MIT, Visiting Professor, 2018, full-time
University of Washington, Warren Francis and Wilma Kolm Bradley Chair,
2014-pres., full-time
University of Washington, Robert E. Dinning Professor, 2009-2014, 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 – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

Asta Networks, Co-founder and interim CEO/CTO, 2000-01
Digital Equipment Corporation, Research Intern, 1990

4. Certifications or professional registrations

5. Current membership in professional organizations

National Academy of Engineering, Member
American Academy of Arts and Sciences, Member
Washington State Academy of Arts and Sciences, Member
ACM, Fellow

6. Honors and awards

National Academy of Engineering
American Academy of Arts and Sciences
Washington State Academy of Arts and Sciences
UNIX Lifetime Achievement Award
UNIX Software Tools and User Group Award
ACM SIGOPS Hall of Fame Award
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 (within and outside of the institution)

Peer Committee Vice Chair, National Academy of Engineering Section 5, 2019
Peer Committee Member, National Academy of Engineering, Section 5, 2017-2018
Chair, University of Washington College of Engineering Endowments Evaluation Committee, 2015 - 2018.
Co-Founder and Steering Committee Chair, PlanetLab Consortium, 2003 - pres.
Co-Founder and Steering Committee Co-Chair, ACM/USENIX Symposium on Network Systems Design and Implementation (NSDI), 2002 - 2012.
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
Guest Editor, Special Issue of IEEE MICRO on Hot Interconnects, February 1996
Program Committee Co-Chair, 1995 IEEE Hot Interconnects III Conference, August 1995
Member, Planning Committee, NSF GENI Project, 2005-07
Associate Editor, ACM Transactions on Computer Systems, 1998-2001
Program Committee Member, ACM SIGCOMM 2000, 2001, 2002, 2004
Program Committee Member, Third Symposium on Operating System Design and Implementation (OSDI), 1999
Program Committee Member, Fourth High Performance Computer Architecture Conference (HPCA-4), 1998
Program Committee Member, First USENIX NT Symposium, 1997
Program Committee Member, Seventh ACM Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-VII), October 1996
Program Committee Member, 1994 and 1995 ACM SIGMETRICS Conferences on the Measurement and Modeling of Computer Systems
Program Committee Member, Fourteenth ACM Symposium on Operating System Principles (SOSP), December 1993.

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

T. Anderson and M. Dahlin. Operating Systems: Principles and Practice, second edition. Recursive Books, 2014. In use at Berkeley, CMU, Cornell, ETH, Maryland, Michigan, NYU, Ohio, Penn State, Princeton, Stanford, Stony Brook, Texas, Texas A&M, UBC, UCF, UCR, UCSB, USC, and over fifty other institutions. See: <http://osp.cs.washington.edu/> for instructor materials.

Q. Zhang, G. Yu, C. Guo, Y. Dang, N. Swanson, X. Yang, R. Yao, M. Chintalapati, A. Refereed Publications Krishnamurthy, T. Anderson. "Deepview: Virtual Disk Failure Diagnosis and Pattern Detection for Azure." Proc. of the Fifteenth USENIX Symposium on Networked Systems Design and Implementation (NSDI), April 2018, pp. 519-532.

Y. Kwon, H. Fingler, T. Hunt, S. Peter, E. Witchel, and T. Anderson. "Strata: A Cross Media File System." Proc. of the Twenty-Sixth Symposium on Operating Systems Principles (SOSP), October 2017.

D. Zhuo, M. Ghobadi, R. Mahajan, K. Forster, A. Krishnamurthy, and T. Anderson. “Understanding and Mitigating Packet Corruption in Data Center Networks.” Proc. of the ACM SIGCOMM 2017 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 2017, pp. 362-375.

N. Kr. Sharma, A. Kaufmann, T. Anderson, C. Kim, A. Krishnamurthy, J. Nelson, and S. Peter. “Evaluating the Power of Flexible Packet Processing for Network Resource Allocation.” Proc. of the Fourteenth USENIX Symposium on Networked Systems Design and Implementation (NSDI), April 2017, pp. 67-82.

D. Zhuo, M. Ghobadi, R. Mahajan, A. Phanishayee, X. Zou, H. Guan, A. Krishnamurthy, and T. Anderson. “RAIL: A Case for Redundant Arrays of Inexpensive Links in Data Center Networks.” Proc. of the Fourteenth USENIX Symposium on Networked Systems Design and Implementation (NSDI), April 2017, pp. 561-576.

R. Cheng, W. Scott, P. Ellenbogen, J. Howell, F. Roesner, A. Krishnamurthy, and T. Anderson. “Radiatus: a Shared-Nothing Server-Side Web Architecture.” Proc. of the ACM Symposium on Cloud Computing 2016 (SoCC), October 2016, pp. 237-250.

D. Zhuo, Q. Zhang, V. Liu, A. Krishnamurthy, and T. Anderson. “Rack-level Congestion Control.” Proc. Fifteenth ACM Workshop on Hot Topics in Networks (HotNets XV), November 2016, pp. 148-154.

S. Peter, J. Li, I. Zhang, D. Ports, D. Woos, A. Krishnamurthy, T. Anderson, and T. Roscoe. “Arrakis: The Operating System is the Control Plane.” ACM Transactions on Computer Systems, 33(4):11, 2016, pp. 1-30. Also selected as Best Paper in Proc. of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI), October 2014, pp. 1-16.

W. Scott, T. Anderson, T. Kohno, and A. Krishnamurthy. “Satellite: Joint Analysis of CDNs and Network-Level Interference.” Proc. of the USENIX Annual Technical Conference (ATC), July 2016, pp. 195-208.

S. Han, H. Shen, T. Kim, A. Krishnamurthy, T. Anderson, and D. Wetherall. “Meta-Sync: Coordinating Storage Across Multiple File Synchronization Services.” IEEE Internet Computing, vol. 20, no. 3, May-June 2016, pp. 36-44.

A. Kaufmann, N. Kr. Sharma, S. Peter, A. Krishnamurthy, and T. Anderson. “High Performance Packet Processing with FlexNIC.” Proc. of the ACM Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 2016, pp. 67-81.

D. Woos, J. Wilcox, S. Anton, Z. Tatlock, M. Ernst, and T. Anderson. “Planning for Change in a Formal Verification of the Raft Consensus Protocol.” Proc. of the ACM SIGPLAN Conference on Certified Programs and Proofs (CPP), January 2016, pp. 154-165.

V. Liu, D. Zhuo, S. Peter, A. Krishnamurthy, and T. Anderson. “Subways: A Case for Redundant, Inexpensive Data Center Edge Links.” Proc. of the International Conference on Emerging Networking Experiments and Technologies (CoNEXT), December 2015, pp. 1-13.

S. Han, H. Shen, T. Kim, A. Krishnamurthy, T. Anderson, and D. Wetherall. “Meta-Sync: File Synchronization Across Multiple Untrusted Storage Services.” Proc. of the USENIX Annual Technical Conference (ATC), June 2015, pp. 83-95.

J. Wilcox, D. Woos, P. Panchekha, Z. Tatlock, X. Wang, M. Ernst, and T. Anderson. “Verdi: A Framework for Implementing and Formally Verifying Distributed Systems.” Proc. of the ACM Symposium on Programming Language Design and Implementation (PLDI), June 2015, pp. 357-368.

A. Kaufmann, S. Peter, T. Anderson, and A. Krishnamurthy. “FlexNIC: Rethinking Network DMA.” Proc. of the 15th USENIX Workshop on Hot Topics in Operating Systems (HotOS), May 2015.

S. Peter, U. Javed, Q. Zhang, D. Woos, A. Krishnamurthy, and T. Anderson. “One Tunnel is (Often) Enough.” Proc. of the ACM SIGCOMM 2014 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 2014, pp. 99-110.

D. Zhuo, Q. Zhang, D. Ports, A. Krishnamurthy and T. Anderson. “Machine Fault Tolerance for Reliable Datacenter Systems.” Proc. of the 5th Asia-Pacific Workshop on Systems (APSys 2014), June 2014.

S. Peter, I. Zhang, D. Ports, J. Li, D. Woos, T. Anderson, A. Krishnamurthy, and M. Zbikowski. “Towards High-Performance Application-Level Storage Management.” Proc. of the 6th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage), June 2014.

S. Han, V. Liu, Q. Pu, S. Peter, T. Anderson, A. Krishnamurthy, and D. Wetherall. “Expressive Privacy Control with Pseudonyms.” Proc. of the ACM SIGCOMM 2013 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication, August 2013, pp. 291-302.

U. Javed, . Cunha, D. Choffnes, E. Katz-Bassett, T. Anderson, and A. Krishnamurthy. “PoiRoot: Investigating the Root Cause of Interdomain Path Changes.” Proc. of the ACM SIGCOMM 2013 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication, August 2013, pp. 183-194.

S. Peter and T. Anderson. “Arrakis: A Case for the End of the Empire.” Proc. 14th USENIX Workshop on Hot Topics in Operating Systems (HotOS), May 2013.

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 - 2012
Arrakis, an operating system for high-performance data center applications.
<https://github.com/UWNetworksLab/arrakis>
Freedom/UProxy, a Javascript plug-in enabling web browsers to reach censored web pages.
<https://github.com/UWNetworksLab/freedom>
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

Balazinska , Magdalena

1. Education – degree, discipline, institution, year

PhD in Computer Science, Massachusetts Institute of Technology, 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 – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

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 – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

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):
Guest lecturer in freshman seminar entitled “Exploring Technology Design Majors”. 02/2013.
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.

Beame, Paul

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
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:
ACM Council 2014-2018
Past Chair, ACM SIGACT 2015-2018
Chair, ACM SIGACT, 2012- 2015,
Chair, IEEE Computer Society TCMF, 2007-11
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
of Washington Service:
Associate Director for Facilities, Allen School for CSE, 2017-present
Associate Chair for Facilities, CSE 2015-2017
for Education, CSE, 2009-2010, 2011-2015
CSE Space Committee (Chair) 2002-10, 2011-Present
CSE Faculty Program Advisor BS/MS Program 2008-10
8. Selected Publications:

Member,

University

Associate Chair

Paul Beame, Vincent Liew. Towards Verifying Nonlinear Integer Arithmetic. In *Proceedings of the International Conference on Computer-Aided Verification (CAV 2017)*, 238-258, July 2017. Also to appear in *Journal of the ACM* 66 (5), 2019.

Paul Beame, Shayan Oveis Gharan, Xin Yang. Time-Space Tradeoffs for Learning Finite Functions from Random Evaluations, with Applications to Polynomials. In *Proceedings of the Conference on Learning Theory (COLT 2018)*, pages 843-856, 2018.

Paul Beame, Paraschos Koutris, and Dan Suciu. Communication Steps for Parallel Query Processing, *Journal of the ACM* 64(6): pages 40:1-40:58, 2017.

Paul Beame, Jerry Li, Sudeepa Roy, Dan Suciu. Exact Model Counting of Query Expressions: Limitations of Propositional Methods. *ACM Transactions on Database Systems* 42(1): 1:1-1:46 (2017)

Paul Beame, Cyrus Rashtchian. Massively-Parallel Similarity Join, Edge-Isoperimetry, and Distance Correlations on the Hypercube. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017)*, pages 289-306, 2017.

Paul Beame, Chris Beck, Russell Impagliazzo. Time-Space Trade-offs in Resolution: Superpolynomial Lower Bounds for Superlinear Space. *SIAM Journal on Computing*. 45(4): pages 1612-1645 (2016)

Paul Beame and Widad Machmouchi.. The quantum query complexity of AC^0 . *Quantum Information & Computation*, 12(7–8):670–676, 2012.

Bricker, Lauren

1. Education

PhD, Computer Science and Engineering, University of Washington, 1998
M.Sc, Computer Science and Engineering, University of Washington, 1993
B.S. Theoretical Mathematics and Pre-Medicine, University of Michigan, 1985

2. Academic experience –

University of Washington, Paul G Allen School of Computer Sci. & Eng., Lecturer and K-12 outreach
2017 –

Lakeside school, Seattle, WA
Faculty 2007-2017

3. Non-academic experience –

Code.org, Seattle, WA (PT)
Computer Science Discoveries Curriculum cohort writer 2016
Teacher trainer/Facilitator/Interviewer 2015 -

Secular Jewish Circle (PT)
Volunteer positions 1997-2014
Teacher 2005-2007
Interim Administrative Assistant 2005-2007

Performant/Mercury Interactive 2001-2003
Program/Project Manager, Quality Assurance Manager, Lead UI Architect. (PT, then FT)

Brickware/Self Employed (PT/FT) 1988 –

MathSoft
Research Engineer (FT) 1999

Ark Interface
Senior Software Engineer (FT) 1994-1997

Adonis Corp/ConnectSoft
Software Engineer (PT/FT) 1990-1993

Caddex Corporation
Software Engineer (FT) 1987-1988

The Boeing Company
Software Engineer (FT) 1985 - 1987

4. Certifications or professional registrations

5. Current membership in professional organizations

- Association for Computing Machinery (ACM)
 - Special Interest Group on Computer Science Education (SIGCSE)
 - Special Interest Group on Computer Graphics (SIGGRAPH)

- Computer Science Teachers Association (CSTA)+ and Puget Sound Computer Science Teachers Association (PSCSTA)
6. Honors and awards
- Influential secondary school faculty chosen to attend a Luncheon with a Stanford Terman Award winner.
 - University of Washington Department of Computer Science and Engineering Inspirational Teacher 2012, 2014
 - GeekWire Geek of the Week. January 5, 2012
7. Service activities (within and outside of the institution)
- Member of the Allen School Diversity Committee, 2017-present.
 - Treasurer, Puget Sound Computer Science Teachers Association 2012 – present.
 - PSCSTA contest organization 2011 -
 - President, Two Herons Foundation 2011 -
 - Moderator for the Freecycle Seattle email list 2004-2008.
 - Volunteer in charge of finance and procurement, Wedgwood Montessori School Fundraiser, 2004.
 - Volunteer, Inquiry-based Science program, Bryant and Lowell Elementary schools, Seattle, WA.
 - CS women's group coordinator, University of Washington, Department of Computer Science and Engineering, 1994-1997.
 - Member of the Demos Committee, User Interface Software and Technology (UIST) conference, 1998.
 - Reviewer, Conference on Human Factors in Computing Systems (CHI), 1998; User Interface Software and Technology, 2002, 2003; IEEE Symposium on Visual Languages, 1996, 1997; and International Journal of Software Engineering and Knowledge Engineering.
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
- Bricker L, Davidson, A. Human Centered Design Charrette. To be presented at the Computer Science Teachers' Association 2019 Annual Conference, July 7-10, 2019, Phoenix, AZ.
 - Bricker, L, Fox, J, Katagiri, C, Locke, D, Phillips, T, Thomas, J, Casteneda, L. Teaching the Maker Mentality: A Conversation about Project Based Learning in K-12 Education, a panel at the Seattle Mini Maker Faire, (Seattle, WA, September 18, 2016).
 - Bricker, L, Sconyers T. Totally Tubular Track: An interactive, collaborative exhibit using low fidelity and upcycled prototyping materials that will result in a marble track on a wall at the Seattle Mini Maker Faire, (Seattle, WA, September 17-18, 2016).
 - Bricker, L. Generating Geeks: Why everyone should learn computational thinking in grade school, how we as technologists can support that learning, and my path to becoming a geek generator at Lakeside School in Seattle. Fremont Women in Tech talk (Seattle, WA, June 21, 2016).
9. Briefly list the most recent professional development activities
- Code.org Facilitator Summits 2016 -
 - Consortium for the Advancement of Undergraduate STEM Education (CAUSE) project meetings 2018-2019
 - UW Center for Teaching and Learning Evidence Based Teaching cohorts (Exploration and Implementation) 2017-2018
 - Attending SIGCSE 2017-
 - Attending SIGGRAPH 2011-
 - Attending Grace Hopper Conference 2011

Ceze, Luis

1. Education – degree, discipline, institution, year

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 – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

Computer Science and Engineering, University of Washington Seattle, WA. Associate Professor, September 2012-present. Assistant Professor, September 2007-August 2012.

3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

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

Member of 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 (within and outside of the institution)

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. 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
 - 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 ^[1]~~SEP~~ *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

Choi, Yejin

1. Education – degree, discipline, institution, year
PhD, Computer Science, Cornell University, 2010
2. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

University of Washington, Associate Professor, Sep 2017 - Present, Full time
University of Washington, Assistant Professor, Sep 2014 - Aug 2017, Full time
Stony Brook University, Assistant Professor, Sep 2010 - Aug 2014, Full time

3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

Allen Institute of Artificial Intelligence, Research Scientist, Jan 2019 - Apr 2019, Part time
Allen Institute of Artificial Intelligence, Research Manager, May 2019 - Present, Part time

4. Certifications or professional registrations
5. Current membership in professional organizations

Member of ACL, IEEE

6. Honors and awards

1. A recipient of Borg Early Career Award (BECA), 2018
2. Alexa Prize winner (UW Sounding Board), 2017
3. Facebook ParlAI Award, 2017
4. AI2 Key Scientific Challenge Award, 2017
5. CACM Research Highlights on “Learning to Name Objects”, 2016
6. IEEE AI’s 10 to Watch, 2015
7. Marr Prize (Best Paper Award) at ICCV 2013
8. Best Presentation Award at KOCSEA Symposium 2013
9. Best Paper Nomination at ICWSM 2012

7. Service activities (within and outside of the institution)

University Service

1. Admission Committee, 2016 – 2018
2. Chair for the CSE Visit Day (recruiting event for incoming PhD), 2018
3. Committee for the CSE Visit Day (recruiting event for incoming PhD), 2015
4. Mentor for the NSF STARS program (in support of economic diversity), 2016 – 2017
5. Lecturer on Conversational AI at UW Math Academy (K-12 program), 2017
6. CSE Women’s Welcome Day, Organizing and Hosting, 2016
7. CSE Women’s Day, Presentation and Discussion, 2015

Professional Society and Other Service

1. Conference Officer at the ACL Exec Board (2015 - 2018)
2. Area Chair: EMNLP (2014, 2012); NAACL (2012)
3. Associate Editor: JAIR (2017 - 2020)

4. Editorial Board: JAIR (2016 - 2017)

International, National, or Governmental Service

1. Panel reviewer for NSF (2011, 2012, 2014, 2016, 2017, 2019)
2. Local host for North American Computational Linguistics Olympiad (NACLO 2012 — 2014)

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

Aug 2019	Talk at <u>NLP for Conversational AI Workshop @ ACL</u>
July 2019	Panel at <u>MSR Faculty Summit on the Future of Work</u> , Keynote at KCCV
Jun 2019	Talk at <u>NeuralGen Workshop @ NAACL</u> , <i>"The Enigma of Neural Text Degeneration as the First Defense to Neural Fake News."</i>
May 2019	Talk at <u>AKBC</u> , Elemental Cognition, IBM Research, <i>"From Atomic to Comet: Commonsense AKBC."</i>
July 2018	Talk at <u>ACL Workshop on Representation Learning for NLP</u>
July 2018	Talk at <u>LxMLS Lisbon Machine Learning School</u>
Jun 2018	Talk at <u>CVPR Workshop on Visual Question Answering</u>
Jun 2018	Talk at <u>NAACL Workshop on Generalization in Deep Learning</u> , <i>"Why NLU Doesn't Generalize Well to NLG."</i>
Apr 2018	Talk at <u>NW-NLP</u> , video available here
Feb 2018	Talk at <u>AAAI</u>

Defending Against Neural Fake News

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, Yejin Choi
arXiv:1905.12616, 2019

*Project Page & Grover Demo

COMET: Commonsense Transformers for Knowledge Graph Construction

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz and Yejin Choi
Association for Computational Linguistics (ACL), 2019

*Demo

Hellaswag: Can a Machine Really Finish Your Sentence?

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi and Yejin Choi
Association for Computational Linguistics (ACL), 2019

The Risk of Racial Bias in Hate Speech Detection

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi and Noah A. Smith
Association for Computational Linguistics (ACL), 2019

From Recognition to Cognition: Visual Commonsense Reasoning (VCR)

Rowan Zellers, Yonatan Bisk, Ali Farhadi, Yejin Choi.

Conference on Computer Vision and Pattern Recognition (CVPR), 2019

*Project Page

ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith and Yejin Choi

Association for the Advancement of Artificial Intelligence (AAAI), 2019.

[*Demo at AI2](#), [*Project Page at UW](#)

SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference

Rowan Zellers, Yonatan Bisk, Roy Schwartz and Yejin Choi

Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018.

[*Project Page](#) [*Leaderboard](#)

Modeling Naive Psychology of Characters in Simple Commonsense Stories

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight and Yejin Choi

Association for Computational Linguistics (ACL), 2018.

[*Project Page \(with an online dataset browser!\)](#)

Learning to Write with Cooperative Discriminators

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub and Yejin Choi

Association for Computational Linguistics (ACL), 2018.

Event2Mind: Commonsense Inference on Events, Intents, and Reactions

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith and Yejin Choi

Association for Computational Linguistics (ACL), 2018.

[*Project Page \(with an online dataset browser!\)](#)

Discourse-Aware Neural Rewards for Coherent Text Generation

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-sen Huang, and Yejin Choi

North American Chapter of the Association for Computational Linguistics (NAACL), 2018.

Deep Communicating Agents for Abstractive Summarization

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi

North American Chapter of the Association for Computational Linguistics (NAACL), 2018.

Neural Poetry Translation

Marjan Ghazvininejad, Yejin Choi, and Kevin Knight

North American Chapter of the Association for Computational Linguistics (NAACL), short paper, 2018.

Neural Motifs: Scene Graph Parsing with Global Context

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi

Conference on Computer Vision and Pattern Recognition (CVPR), 2018

Simulating Action Dynamics with Neural Process Networks

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi

International Conference on Learning Representations (ICLR), 2018.

[\[Podcast\]](#)

Verb Physics: Relative Physical Knowledge of Actions and Objects

Maxwell Forbes and Yejin Choi

Association for Computational Linguistics (ACL), 2017.

[*Project Page](#)

Connotation Frames: A Data-Driven Investigation

Hannah Rashkin, Sameer Singh, and Yejin Choi
Association for Computational Linguistics (ACL), 2016.
*Project Page

9. Briefly list the most recent professional development activities

Curless, Brian L.

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

Academic experience

Director, UW Reality Lab, Paul G. Allen School of Computer Science & Engineering (2018 – present).

Professor, University of Washington, Paul G. Allen School 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).

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).

Professional organizations

Association for Computing Machinery (1998-present)

IEEE Computer Society (1998-present)

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)

Selected Service Activities

Postdoc advisor, UW CSE, (2015-present).

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), 2012-1013).

Executive Committee, UW CSE (2003-2004, 2010-2011).

Chair Search Committee, UW CSE (2000-2001).

Selected Recent Publications

Chung-Yi Zheng, Brian Curless, Ira Kemelmacher-Shlizerman. Photo Wake-up: 3D Character Animation from a Single Photo. *Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Konstantinos Rematas, Ira Kemelmacher-Shlizerman, Brian Curless, Steve Seitz. Soccer on Your Tabletop. *Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Edward Zhang, Michael Cohen, and Brian Curless. Discovering Point Lights with Intensity Distance Fields. *Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Edward Zhang, Michael F. Cohen, and Brian Curless. Emptying, Refurnishing, and Relighting Indoor Spaces. *SIGGRAPH Asia / ACM Transactions on Graphics*, December 2016.

Juliet Fiss, Brian Curless, Rick Szeliski. Light Field Layer Matting. *Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Qi Shan, Changchang Wu, Brian Curless, Yasutaka Furukawa, Carlos Hernandez and Steven M. Seitz. Accurate Geo-registration by Ground-to-Aerial Image Matching. *3DV (3D Vision) 2014*, December 2014.

Qi Shan, Brian Curless, Yasutaka Furukawa, Carlos Hernandez and Steven M. Seitz, Photo Uncrop. *Proceedings of European Conference on Computer Vision (ECCV)*, September 2014.

Qi Shan, Brian Curless, Yasutaka Furukawa, Carlos Hernandez and Steven M. Seitz, Occluding Contours for Multi-View Stereo. *Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Ankit Gupta, Maneesh Agrawala, Brian Curless, and Michael F. Cohen. MotionMontage: A System to Annotate and Composite Motion Takes for 3D Animations. *CHI 2014*, April 2014.

Domingos, Pedro

1. Education – degree, discipline, institution, year
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 – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time
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 – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time
4. Certifications or professional registrations
5. Current membership in professional organizations
ACM, ACM SIGKDD, AAAI, IMLS.
6. Honors and awards
Fellowships: AAAI, 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 (within and outside of the institution)
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 AAAI, 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

Farhadi, Ali

1. Education – degree, discipline, institution, year
PhD. Computer Science, University of Illinois at Urbana-Champaign, 2011
2. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

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 – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

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 (within and outside of the institution)

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. 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
 - 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. Briefly list the most recent professional development activities

Fogarty, James

1. Education:
Ph.D. Human Computer Interaction, Carnegie Mellon University, 2006
B.S. Computer Science, Virginia Tech, 2000
2. Academic Experience:
Professor, Computer Science & Engineering, U. Washington, 2017-Present
Associate Professor, Computer Science & Engineering, U. Washington, 2011-2017
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
University of Washington Research Innovation Award, 2015
Carnegie Mellon University Allen Newell Award for Research Excellence, 2015
Best Paper Honorable Mention, ASSETS 2018
Best Paper Honorable Mention, DIS 2018
Best Paper Honorable Mention, ASSETS 2017
Best Paper Award, CHI 2017
Best Paper Honorable Mention, CHI 2017
Best Paper Award, CSCW 2016
Best Paper Honorable Mention, CHI 2015
Best Paper Honorable Mention, CHI 2014
Best Paper Award, dg.o 2011
Best Paper Award, CHI 2010
Best Paper Honorable Mention, UbiComp 2009
Best Paper Honorable Mention, CHI 2009
Best Paper Award, CHI 2005
7. Selected Service Activities:
National and International:
Co-Chair, CHI 2017 Subcommittee on Engineering Interactive Systems and Technologies
Co-Chair, CHI 2012 Subcommittee on Technology, Systems, and Tools
Program Committee, CHI 2008, 2009, 2010, 2012, 2013, 2015, 2016, 2017, 2020
Program Committee, UIST 2006, 2008, 2010, 2016
Governing Board Member, Human Computer Interaction Consortium, 2008-present
Chair, IJCAI 2009 Workshop on Intelligence and Interaction

University of Washington Service:
Director, DUB Interdisciplinary Committee, 2009/10-present

UW Population Health Executive Committee, 2016/17-2018/19
UW CSE Graduate Admissions and Recruiting HCI Area Organizer, 2009/10-2012/13
UW CSE Undergraduate Admissions Committee, 2010/11

8. Selected Publications:

Pina L.R., Sien S.-W., Ward T., Yip J.C., Munson S.A., Fogarty J., Kientz J.A. (2017). From Personal Informatics to Family Informatics: Understanding Family Practices around Health Monitoring. Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2017), pp. 2300-2315.

Chung C.F., Dew K., Cole A., Zia J., Fogarty J., Kientz J.A., Munson S.A. (2016). Boundary Negotiating Artifacts in Personal Informatics: Patient-Provider Collaboration with Patient Generated Data. Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2016). pp. 768-784. (Best Paper Award)

Cordeiro F., Epstein D.A., Thomaz E., Bales E., Jagannathan A.K., Abowd G., Fogarty J. (2015). Barriers and Negative Nudges: Exploring Challenges in Food Journaling. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2015). pp. 1159-1162.

Epstein D.A., Ping A., Fogarty J., Munson S.A. (2015). A Lived Informatics Model of Personal Informatics. Proceedings of the International Conference on Pervasive and Ubiquitous Computing (UbiComp 2015). pp. 731-742.

Schroeder J., Chung C., Epstein D.A., Karkar R., Parsons A., Murinova N., Fogarty J., and Munson S.A. Examining Self-Tracking by People with Migraine: Goals, Needs, and Opportunities in a Chronic Health Condition. Proceedings of the ACM Conference on Designing Interactive Systems (DIS 2018), pp. 135-148.

Karkar R., Schroeder J., Epstein D.A., Pina L.R., Scofield J., Fogarty J., Kientz J.A., Munson S.A., Vilardaga R., Zia J. (2017). TummyTrials: A Feasibility Study of Using Self-Experimentation to Detect Individualized Food Triggers. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2017), pp. 6850-6863.

Schroeder J., Hoffswell J., Chung C., Fogarty J., Munson S., Zia J. (2017). Supporting Patient-Provider Collaboration to Identify Individual Triggers using Food and Symptom Journals. Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2017), pp. 1726-1739.

Karkar R., Zia J., Vilardaga R., Mishra S.R., Fogarty J., Munson S.A., Kientz J.A. (2015). A Framework for Self-Experimentation in Personalized Health. Journal of the American Medical Informatics Association (JAMIA). ocv150.

Choe E.K., Abdullah S., Rabbi M., Thomaz E., Epstein D.A., Cordeiro F., Kay M., Abowd G.D., Choudhury T., Fogarty J., Lee B., Matthews M., Kientz J.A. (2017). Semi-Automated Tracking: A Balanced Approach for Self-Monitoring Applications. IEEE Pervasive Computing. 16(1), pp. 74-84.

Zhang X., Pina L.R., Fogarty J. (2016). Examining Unlock Journaling with Diaries and Reminders for In-Situ Self Report in Health and Wellness. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2016). pp. 5658-5664. PMID: PMC5553980.

Epstein D.A., Cordeiro F., Fogarty J., Hsieh G., Munson S.A. (2016). Crumbs: Lightweight Daily Food Challenges to Promote Engagement and Mindfulness. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2016). pp. 5632-5644.

Cordeiro F., Bales E., Cherry E., Fogarty J. (2015). Rethinking the Mobile Food Journal: Exploring Opportunities for Lightweight Photo-Based Capture. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2015). pp. 3207-3216.

Patel K., Bancroft N., Drucker S., Fogarty J., Ko A.J., Landay J.A. Gestalt: Integrated Support for Implementation and Analysis in Machine Learning Processes. (2010). Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2010), pp. 37-46.

Hoffmann R., Amershi S., Patel K., Wu F., Fogarty J., Weld D.S. (2009). Amplifying Community Content Creation with Mixed Initiative Information Extraction. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2009), pp. 1849-1858.

Fogarty J., Au C., Hudson S.E. (2006). Sensing from the Basement: A Feasibility Study of Unobtrusive and Low-Cost Home Activity Recognition. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2006), pp. 91-100.

Fogarty J., Hudson S.E., Atkeson C.G., Avrahami D., Forlizzi J., Kiesler S., Lee J.C., Yang J. (2005). Predicting Human Interruptibility with Sensors. ACM Transactions on Computer Human Interaction (TOCHI), Vol. 12, No. 1, March 2005, pp. 119-146.

Zhang X., Ross A.S., Caspi A., Fogarty J., Wobbrock J.O. (2017). Interaction Proxies for Runtime Repair and Enhancement of Mobile Application Accessibility. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2017), pp. 6024-6037.

Dixon M., Nied A.C., Fogarty J. (2014). Prefab Layers and Annotations: Extensible Pixel-Based Interpretation of Graphical Interfaces. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2014). pp. 221-230.

Dixon M., Leventhal D., Fogarty J. (2011). Content and Hierarchy in Pixel-Based Methods for Reverse Engineering Interface Structure. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2011). pp. 969-978.

Dixon M., Fogarty J. (2010). Prefab: Implementing Advanced Behaviors Using Pixel-Based Reverse Engineering of Interface Structure. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2010). pp. 1525-1534.

9. Professional Development Activities:

Fox, Dieter

1. Education – degree, discipline, institution, year
 - 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 – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time
 - Associate Professor, ^[SEP]University of Washington, Department of Computer Science & Engineering: 2005 – present.
 - Adjunct Associate Professor, ^[SEP]University of Washington, Department of Electrical Engineering: 2005 – present.
 - Assistant Professor^[SEP]University of Washington, Department of Computer Science & Engineering: 2000 – 2005.
 - Adjunct Assistant Professor^[SEP]University of Washington, Department of Electrical Engineering: 2000 – 2005.
 - Postdoctoral research associate,^[SEP]Carnegie Mellon University, School of Computer Science: 1998 – 2000.
3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time
 - Director, Intel Labs Seattle, Seattle, WA: 2009 – 2011.
4. Certifications or professional registrations
5. Current membership in professional organizations
 - IEEE, ACM, AAAI.
6. Honors and awards
 - IEEE Senior Member, 2012.
 - AIJ Prominent Paper Award: Artificial Intelligence Journal, 2012.
 - Fellow of the AAAI: Association for the Advancement of Artificial Intelligence, 2011.
 - AAAI 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 (within and outside of the institution)
 - 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. 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. 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.

Froehlich, Jon E.

1. Education – degree, discipline, institution, year
PhD, Computer Science and Engineering, UW, 2011
MS, Computer Science and Engineering, UW, 2006
MS, Information and Computer Science, UC Irvine, 2004
BS, Computer Engineering, Iowa State, 2001
2. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time
Associate Professor, Allen School of CSE, UW, 2017-Present
Assistant Professor, Computer Science, University of Maryland, 2012-2017
3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time
Telefonica Research, Research Intern, Summer 2008
Microsoft Research, Research Intern, Summer 2007
Intel Research, Research Intern, Summer 2005, Summer/Fall 2006
4. Certifications or professional registrations
5. Current membership in professional organizations
Association for Computing Machinery (ACM)
6. **Honors and awards**
Best Paper Award, CHI 2019
Best Paper Award, CHI 2019
10-Year Impact Award, UbiComp 2018
Sloan Fellow, 2017
NSF CAREER Award, 2017
Google Faculty Award, 2017
Best Paper Award, CHI 2017
Best Late-Breaking Work, CHI 2016
ACM Computing Review “Best of Computing 2014”
Best Paper Honorable Mention, CHI 2015
Best Paper Honorable Mention, CHI 2015
Best Paper Award, ASSETS 2013
Best Paper Award, CHI 2013
Best Paper Honorable Mention, CHI 2013
Google Faculty Award, 2012
Best Paper Honorable Mention, CHI 2012
UW Graduate School Distinguished Dissertation Award, 2012
UW Allen School William Chan Memorial Dissertation Award, 2012
UW College of Engineering Student Innovator Award for Research, 2011
Best Paper Award, CHI 2010
Best Paper Honorable Mention, UbiComp 2009
UW Environmental Challenge Winner, 2009
Madrona Prize for Research Excellence, 2009
Microsoft Research Fellowship, 2008

7. Service activities (within and outside of the institution)
 - Associate Chair, ACM Intl SIGACCESS Conference on Computers and Accessibility (2016, 2017, 2018)
 - Associate Chair, ACM Intl SIGACCESS Conference on Computers and
 - Subcommittee Chair, ACM Human Factors in Computing Systems (CHI'15)
 - Associate Chair, ACM Intl SIGACCESS Conference on Computers and Accessibility (ASSETS'15)
 - Subcommittee Chair, ACM Human Factors in Computing Systems (CHI'14)
 - Associate Chair, ACM Intl Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'14)
 - Associate Chair, ACM Human Factors in Computing Systems (CHI'13)
 - Associate Chair, International Conference on Pervasive Computing, 2011
 - Co-Chair of Doctoral Consortium with Amanda Lazar, ASSETS 2019
 - Co-Chair of Family and Childcare Support with Audrey Girouard, CHI 2018
 - Chair of Best of CHI Awards, CHI 2016
 - Assistant to Chair at ACM Human Factors in Computing Systems (CHI), 2008

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**
 - Weld, G., Jang, E., Zeng, A., Li, A., Heimerl, K., **Froehlich, J.** (2019) “Deep Learning for Automatically Detecting Sidewalk Accessibility Problems Using Streetscape Imagery.” *In submission*

 - Jain, D., Desjardins, A., Findlater, L., **Froehlich, J.** (2019) “Autoethnography of a Hard of Hearing Traveler.” *In submission*.

 - He, L., Peng, H., Lin, M., Konjeti, R., Guimbretiere, F., **Froehlich, J.** (2019) “Ondulé: Designing and Controlling 3D Printable Springs.” *In submission*.

 - Kang, S., Norooz, L., Bonsignore, E., Byrne, V., Clegg, T. L., **Froehlich, J. E.** (2019). “PrototypAR: Prototyping and Simulating Complex Systems with Paper Craft and Augmented Reality.” Proceedings of IDC 2019, Boise, Idaho, June 6-9, 2019.

 - Saha, M., Saugstad, M., Maddali, H., Zeng, A., Holland, R., Bower, S., Dash, A., Chen, S., Li, A., Hara, K., **Froehlich, J.** “Project Sidewalk: A Web-based Crowdsourcing Tool for Collecting Sidewalk Accessibility Data At Scale” Proceedings of CHI'19, Glasgow, Scotland, May 4-9, 2019. [Acceptance Rate: 23.8% (705/2960); **BEST PAPER WINNER – TOP 1%**].

 - Hiniker, A., **Froehlich, J.**, Zhang, M., Beneteau, E. “Anchored Audio Sampling: A Seamless Method for Exploring Children’s Thoughts and Reactions During Deployment Studies” Proceedings of CHI'19, Glasgow, Scotland, May 4-9, 2019. [Acceptance Rate: 23.8% (705/2960); **BEST PAPER WINNER – TOP 1%**].

 - Mauriello, M., McNally, B., **Froehlich, J.** “Thermporal: An Easy-To-Deploy Temporal Thermographic Sensor System to Support Residential Energy Audits” Proceedings of CHI'19, Glasgow, Scotland, May 4-9, 2019. [Acceptance Rate: 23.8% (705/2960)]

 - Jain, D., Lin, A., Guttman, R., Amalachandran, M., Zeng, A., Findlater, L., **Froehlich, J.** “Exploring Sound Awareness in the Home for People who are Deaf or Hard of Hearing” Proceedings of CHI'19, Glasgow, Scotland, May 4-9, 2019. [Acceptance Rate: 23.8% (705/2960)]

 - Findlater, J., Chinh, B., Jain, D., **Froehlich, J.**, Kushalnagar, R., Lin, A. “Deaf and Hard-of-hearing Individuals’ Preferences for Wearable and Mobile Sound Awareness Technologies. Proceedings of CHI'19, Glasgow, Scotland, May 4-9, 2019. [Acceptance Rate: 23.8% (705/2960)]

Gollakota, Shyamnath

1. Education – degree, discipline, institution, year
Ph.D., Electrical Engineering and Computer Science, MIT, 2013
2. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

University of Washington, Assistant Professor, 2012 - present
3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

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 (within and outside of the institution)

PC Member: 2013: SIGCOMM, MOBICOM, MOBISYS
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
 - 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

Grossman, Dan

1. Education – degree, discipline, institution, year

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 – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

Deputy Director, Paul G. Allen School of Computer Science & Engineering, July 2017–
Associate Chair of Education, Computer Science & Engineering, July 2015–
Acting Chair, Computer Science & Engineering, December 2016– February 2017
University of Washington, J. Ray Bowen Professor of Innovation in Engineering Education, 2013 – 2018
University of Washington, Professor, 2015 –
University of Washington, Associate Professor, 2009 – 2015
University of Washington, Assistant Professor, 2003 – 2009
Harvard University, Visiting Associate Professor, 2011

3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

Microsoft Research, Visiting Researcher, 2010

4. Certifications or professional registrations
5. Current membership in professional organizations

ACM

6. Honors and awards

ACM SIGPLAN Distinguished Service Award, 2015
University of Washington, J. Ray Bowen Professor of Innovation in Engineering Education, 2013 -- 2018
1st Place, UW ACM Teacher of the Year' Award, 2008
2nd Place, UW ACM Teacher of the Year' Award, 2006
1st Place, UW ACM Teacher of the Year' Award, 2005
National Science Foundation CAREER Award, 2005
Intel Graduate Student Fellowship, 2002—2003
National Science Foundation Fellowship, 1998--2001

7. Service activities (within and outside of the institution)
Outside institution:

2018: PLDI: ACM Conference on Programming Language Design and Implementation, Program Chair

2014—2020: CRA Board of Directors
 2016-present: CRA Education Committee
 2015-present: CRA Government Affairs Committee
 2013--2017: ACM Education Board
 2011--2013: ACM/IEEE Computing Curricula 2013: Computer Science,
 Steering Committee
 2010--2017: ACM Education Council
 2009--2012: ACM SIGPLAN Executive Committee
 Over 35 conference program committees including PLDI, POPL, OOPSLA, and ASPLOS
 2009 TRANSACT: 4th ACM SIGPLAN Workshop on Transactional Computing, Program Chair

Inside institution:

Data Science Provost Task Force, 2019
 Enrollment Management Data and Analytics Work Group, 2018
 Taskforce on Budget Supplement Distribution, 2018
 Search Committee for Provost and Executive Vice President, 2017–2018
 UW Leadership Workshop Participant, 2017–2018
 Senate Committee on Planning and Budgeting, 2016–present
 Global Innovation Project Advisory Committee, 2014–2015
 Search committee for Vice Provost of Educational Outreach, 2014
 Online Education Task Force, 2013–2014

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

Sinking Point: Dynamic Precision Tracking for Floating-Point
 Bill Zorn, Dan Grossman, Zach Tatlock.
 Conference for Next Generation Arithmetic, Singapore, March, 2019.

Concerto: A Framework for Combined Concrete and Abstract Interpretation
 John Toman, Dan Grossman.
 ACM Symposium on Principles of Programming Languages, Lisbon, Portugal, January, 2019.

Functional Programming for Compiling and Decompiling Computer-Aided Design
 Chandrakana Nandi, James R. Wilcox, Pavel Panchekha, Taylor Blau, Dan Grossman, Zachary Tatlock.
 23rd International Conference on Functional Programming, St. Louis, MO, September, 2018.

Legato: An At-Most-Once Analysis with Applications to Dynamic Configuration Updates
 John Toman, Dan Grossman.
 European Conference on Object-Oriented Programming, Amsterdam, Netherlands, June, 2018.

Œuf: Minimizing the Coq Extraction TCB
 Eric Mullen, Stuart Pernsteiner, James R. Wilcox, Zachary Tatlock, Dan Grossman.
 The 7th ACM SIGPLAN International Conference on Certified Programs and Proofs, Los Angeles, CA,
 January, 2018.

Adapting Proof Automation to Adapt Proofs
 Talia Ringer, Nathaniel Yazdani, John Leo, Dan Grossman.
 The 7th ACM SIGPLAN International Conference on Certified Programs and Proofs, Los Angeles, CA,
 January, 2018.

Instrumentation Bias for Dynamic Data Race Detection
 Benjamin P. Wood, Man Cao, Michael D. Bond, Dan Grossman.

ACM Conference on Object-Oriented Programming Systems, Languages, and Applications, Vancouver, Canada, October, 2017.

A Solver-Aided Language for Test Input Generation

Talia Ringer, Dan Grossman, Daniel Schwartz-Narbonne, Serdar Tasiran.

ACM Conference on Object-Oriented Programming Systems, Languages, and Applications, Vancouver, Canada, October, 2017.

Solver Aided Reverse Engineering of Architectural Features

Bill Zorn, Dan Grossman, Luis Ceze.

14th Annual Workshop on Duplicating, Deconstructing and Debunking, Toronto, Canada, June, 2017.

Debugging Probabilistic Programs

Chandrakana Nandi, Dan Grossman, Adrian Sampson, Todd Mytkowicz, Kathryn S. McKinley.

The First ACM SIGPLAN Workshop on Machine Learning and Programming Languages, Barcelona, Spain, June, 2017.

Verifying Invariants of Lock-Free Data Structures with Rely-Guarantee and Refinement Types

Colin S. Gordon, Michael D. Ernst, Dan Grossman, and Matthew J. Parkinson.

ACM Transactions on Programming Languages and Systems, 39(3), May, 2017.

Programming Language Tools and Techniques for 3D Printing

Chandrakana Nandi, Anat Caspi, Dan Grossman, Zachary Tatlock.

SNAPL: The 2nd Summit On Advances in Programming Languages, Asilomar, CA, May, 2017.

Taming the Static Analysis Beast

John Toman, Dan Grossman.

SNAPL: The 2nd Summit On Advances in Programming Languages, Asilomar, CA, May, 2017.

AUDACIOUS: User-Driven Access Control with Unmodified Operating Systems

Talia Ringer, Dan Grossman, Franziska Roesner.

23rd ACM Conference on Computer and Communications Security, Vienna, Austria, October, 2016.

Staccato: A Bug-Finder for Dynamic Configuration Updates

John Toman, Dan Grossman.

European Conference on Object-Oriented Programming, Rome, Italy, July, 2016.

Verified Peephole Optimizations for CompCert

Eric Mullen, Daryl Zuniga, Zachary Tatlock, Dan Grossman.

ACM Conference on Programming Language Design and Implementation, Santa Barbara, CA, June, 2016.

Optimizing Synthesis with Metasketches

James Bornholt, Emina Torlak, Dan Grossman, Luis Ceze.

ACM Symposium on Principles of Programming Languages, St. Petersburg, FL, January, 2016.

Probability Type Inference for Flexible Approximate Programming

Brett Boston, Adrian Sampson, Dan Grossman, Luis Ceze.

ACM Conference on Object-Oriented Programming Systems, Languages, and Applications, Pittsburgh, PA, October, 2015.

Approximate Program Synthesis

James Bornholt, Emina Torlak, Luis Ceze, Dan Grossman.

Workshop on Approximate Computing Across the Stack, Portland, OR, June, 2015.

Toward a Dependability Case Language and Workflow for a Radiation Therapy System
Michael D. Ernst, Dan Grossman, Jon Jacky, Calvin Loncaric, Stuart Pernsteiner, Zachary Tatlock, Emina Torlak, Xi Wang.

The Inaugural Summit On Advances in Programming Languages, Asilomar, CA, May, 2015.

Monitoring and Debugging the Quality of Results in Approximate Programs
Michael F. Ringenburt, Adrian Sampson, Isaac Ackerman, Luis Ceze, Dan Grossman.
20th International Conference on Architectural Support for Programming Languages and Operating Systems, Istanbul, Turkey, March, 2015.

Peek: A Formally Verified Peephole Optimization Framework for x86
Eric Mullen, Zachary Tatlock, Dan Grossman.
1st International Workshop on Coq for Programming Languages, Mumbai, India, January, 2015.

Symbolic Execution of Multithreaded Programs from Arbitrary Program Contexts
Tom Bergan, Dan Grossman, Luis Ceze.
ACM Conference on Object-Oriented Programming Systems, Languages, and Applications, Portland, OR, October, 2014.

Support the Data Enthusiast: Challenges for Next-Generation Data-Analysis Systems
Kristi Morton, Magdalena Balazinska, Dan Grossman, Jock Mackinlay.
Proceedings of the 40th International Conference on Very Large Databases (Vision Track), Hangzhou, China, September, 2014.

How Programming Languages Will Co-evolve with Software Engineering: A Bright Decade Ahead
Emerson Murphy-Hill, Dan Grossman.
36th International Conference on Software Engineering — Future of Software Engineering track, Hyderabad, India, June, 2014.

Test-Driven Synthesis
Daniel Perelman, Sumit Gulwani, Dan Grossman, Peter Provost.
ACM Conference on Programming Language Design and Implementation, Edinburgh, Scotland, June, 2014.

Expressing and Verifying Probabilistic Assertions
Adrian Sampson, Pavel Panchekha, Todd Mytkowicz, Kathryn S. McKinley, Dan Grossman, Luis Ceze.
ACM Conference on Programming Language Design and Implementation, Edinburgh, Scotland, June, 2014.

9. Briefly list the most recent professional development activities

Heimerl, Kurtis

1. Ph.D., Computer Science, University of California, Berkeley, 2013
M.S., Computer Science, University of California, Berkeley, 2019
B.E., Computer Science & Engineering, University of Washington, 2007
2. Academic experience:
 - Assistant Professor, University of Washington, Computer Science & Engineering, Seattle, WA. (Fall 2016 - present).
3. Non-academic experience:
 - Visiting Researcher, Facebook, Seattle, WA. (2016-2017)
 - Software Engineer, Facebook, Menlo Park, CA. (2015-2016)
 - CEO/Founder, Endaga, Oakland, CA. (2013-2015)
 - Community Manager, Range Networks, San Francisco, CA (2011-2014)
 - Intern at Amazon (Sum 2005, 2006), Google (Sp 2007), Microsoft Research India (Su 2007)
4. Certifications or professional registrations
5. Current membership in professional organizations
 - ACM, USENIX
6. Honors and awards
 - UW College of Engineering “Early Career” Diamond Award 2018
 - MIT Technology Review “35under35” 2014
 - ISIF Asia Community Network Award 2018
 - Community Award at NSDI2019: Scaling Community Cellular Networks with CommunityCellularManager 2019
 - Best Paper at DySPAN2014: GSM Whitespaces: An Opportunity for Rural Cellular Service 2014
 - Community Award at NSDI2013: Expanding Rural Cellular Networks with Virtual Coverage 2013
 - Best Paper at CHI2012: Communitysourcing: Engaging Local Crowds to Perform Expert Work Via Physical Kiosks 2012
 - NSF Graduate Research Fellowship 2008
 - Bob Bandes Memorial Teaching Award (Honorable Mention) 2005
7. Service activities (within and outside of the institution)
 - Demos Chair: ACM ICTD 2019
 - Notes Chair: ACM COMPASS 2018
 - Program Committee: ACM CHI 2020
 - Program Committee: ACM COMPASS 2018,2019
 - Program Committee: ICTD 2013,2015,2016,2017,2019
 - Program Committee: ACM DEV 2014,5
 - Program Committee: HotMobile 2015
 - Program Committee: CHI Work-in-Progress 2013,2014
8. Publications

- Scaling Community Cellular Networks with CommunityCellularManager. Shaddi Hasan, Mary Claire Barela, Matthew Johnson, Eric Brewer, Kurtis Heimerl. 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI) 2019 (Community Award)
- Fresh Insights: User Research Towards a Market Information Service for Bihari Vegetable Farmers. Matthew Ziegler, Lokesh Garg, Shailesh Tiwary, Aditya Vashistha, Kurtis Heimerl. Tenth International Conference on Information and Communication Technologies and Development (ICTD '19) 2019
- Designing Sustainable Rural Infrastructure Through the Lens of OpenCellular. Kashif Ali, Kurtis Heimerl. Communications of the ACM, August 2018
- ThinSIM-based Attacks on Mobile Money Systems Rowan Phipps, Shrirang Mare, Peter Ney, Jennifer Rose Webster, Kurtis Heimerl. First ACM conference on Computing & Sustainable Societies (COMPASS) 2018
- Crowdsourcing Rural Network Maintenance and Repair via Network Messaging. Esther Jang, Mary Claire Barela, Matt Johnson, Philip Martinez, Cedric Festin, Maragaret Lynn, Josephine Dionisio, Kurtis Heimerl. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI) 2018
- An Investigation of Phone Upgrades in Remote Community Cellular Networks. Kushal Shah, Philip Martinez, Emre Tepedelenlioglu, Shaddi Hasan, Cedric Angelo Festin, Joshua Blumenstock, Jo Dionisio, Kurtis Heimerl. International Conference on Information and Communication Technologies and Development (ICTD) 2017
- Towards Building a Community Cellular Network in the Philippines: Initial Site Survey Observations Mary Claire Barela, Mae Sincere Blanco, Philip Martinez, Miguel Carlo Purisima, Kurtis Heimerl, Matthew Podolsky, Eric Brewer, Cedric Angelo Festin. International Conference on Information and Communication Technologies and Development (ICTD) 2016
- A Longitudinal Study of Local, Sustainable, Small-Scale Cellular Networks. Kurtis Heimerl, Shaddi Hasan, Kashif Ali, Eric Brewer, Tapan Parikh. Information Technologies & International Development (ITID) 2015
- Analysis of Smartphone Adoption and Usage in a Rural Community Cellular Network. Kurtis Heimerl, Anuvind Menon, Shaddi Hasan, Kashif Ali, Eric Brewer, Tapan Parikh. International Conference on Information and Communication Technologies and Development (ICTD) 2015
- Rescue Base Station. Ibrahim Ghaznavi, Kurtis Heimerl, Umar Muneer, Abdullah Hamid, Kashif Ali, Tapan Parikh, Umar Saif. 5th Annual ACM Symposium on Computing for Development (DEV) 2014
- GSM Whitespaces: An Opportunity for Rural Cellular Service. Shaddi Hasan, Kurtis Heimerl, Kate Harrison, Kashif Ali, Anant Sahai, Eric Brewer. IEEE Dynamic Spectrum Access Networks (DySPAN) 2014 (Best Paper Award)

9. Briefly list the most recent professional development activities

Hemingway, Bruce

Education – A.B. in Music (*emphasis in Electronic Music Composition and Orchestral Conducting*) Indiana University, 1973.

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.

VR Audio Advisor to the UW Reality Studio, supporting audio spatialisation techniques in undergraduate courses in VR design and programming, 2018-present.

Manager of the Baxter Computer Engineering Laboratory, 2002-present.

Non-academic experience –

Co-founder, CEO and digital designer, dB Technologies, Inc., Seattle, WA. 1993-2002, full time.

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>

Jamieson, Kevin

10. Education – PhD Electrical
and Computer Engineering, University of Wisconsin, Madison, 2015
M.S. Electrical Engineering, Columbia University, 2010
B.S. Electrical Engineering, University of Washington, 2009
11. Academic experience –
Assistant Professor, Computer Science & Engineering, U. Washington, 2017-2019
Postdoctoral Associate, Computer Science, U.C. Berkeley, 2015-2017.
12. Non-academic experience -
Consultant, Optimizely, 2017-2018
Consultant/post-doctoral visitor, Google, 2016-2017
13. Certifications or professional registrations
14. Current membership:
15. Honors and awards
Harold A Peterson Best Dissertation Award, 2016
Sandia National Labs Graduate Research Fellowship, 2013
16. Selected service activities:
Area Chair for ICML 2019, NeurIPS 2019. PC for SysML 2018. Conference reviewer for ICML, NeurIPS, COLT, AISTATS, IJCAI. Journal reviewer for JMLR, INFORMS, IEEE Transactions on Signal Processing, IEEE Transactions on Information Theory
17. Selected Publications:

A Bandit Approach to Multiple Testing with False Discovery Control, Kevin Jamieson, Lalit Jain, Neural Information Processing Systems (NIPS), 2018.

Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization, Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, Ameet Talwalkar, Journal of Machine Learning Research (JMLR), 2018.

Finite Sample Prediction and Recovery Bounds for Ordinal Embedding, Lalit Jain, Kevin Jamieson, Robert Nowak, Neural Information Processing Systems (NIPS), 2016.

NEXT: A System for Real-World Development, Evaluation, and Application of Active Learning, Kevin Jamieson, Lalit Jain, Chris Fernandez, Nick Glattard, Robert Nowak, Neural Information Processing Systems (NIPS), 2015.

lil' UCB : An Optimal Exploration Algorithm for Multi-Armed Bandits, Kevin Jamieson, Matt Malloy, Robert Nowak, and Sebastien Bubeck, Conference on Learning Theory (COLT), 2014.

Karlin, Anna R

Education

1981, B.Sc. in Mathematical Sciences, Stanford University.
1987, Ph.D. in Computer Science, Stanford University.

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

Non-academic experience

1988-94, Principal Scientist (theoretical computer science research), Digital Equipment Corporation's Systems Research Center. Palo Alto, CA.

Current membership in professional organizations

ACM, SIAM

Honors and awards

American Academy of Arts and Sciences Member, elected 2016.
ACM Fellow, 2012.

Selected Plenary and Distinguished Lectures:

14th Conference on Web and Internet Economics, 2018.
Plenary Lecture, SIAM Conference on Discrete Mathematics, 2018.
Computation in Europe 2013: The Nature of Computation: Logic, Algorithms, Applications.
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.

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.

Service activities:

Chair, Faculty recruiting committee, Executive committee, Chair and member of graduate admission committee, Associate Director for Graduate Studies, University of Washington Allen School.
 General Chair, ACM Conference on Economics and Computation (FCRC), 2019.
 Program Chair, 9th Innovations in Theoretical Computer Science, 2018.
 Member, SIGACT Executive Board, 2015-2018.
 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 Kannellakis Theory and Practice Award Subcommittee 2006.
 Program Chair, IEEE Symposium on Foundations of Computer Science, 1997.
 Member of numerous conference program committees.

Selected Publications

A Simply Exponential Upper Bound on the Maximum Number of Stable Matchings, with S. Oveis Gharan and R. Weber, In Proc. of ACM Symposium on Theory of Computation (STOC), 2018.

Stability of Service Under Time-of-Use Pricing, with S. Chawla, N. Devanur, A. Holroyd, J. Martin and B. Sivan In Proc. of ACM Symposium on Theory of Computation (STOC), 2017.

Game Theory, Alive, with Y. Peres. Book published by the American Mathematical Society. May 2017.

The Fedex Problem, with A. Fiat, K. Goldner and E. Koutsoupias, In 17th ACM Conference on Economics and Computation (EC), 2016.

Auctions for Structured Procurement, with M. Cary, A. Flaxman, and J. Hartline, In *Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 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. Immerlica, M. Mahdian and A. Rudra, In *36th Int. Colloquium on Automata, Languages and Programming (ICALP)*, 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.

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.

Krishnamurthy, Arvind

1. Education – degree, discipline, institution, year

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

Professor, Computer Science and Engineering, University of Washington, 2016-now.
Associate Professor, Computer Science and Engineering, University of Washington, 2011-2016.
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

5. Current membership in professional organizations

ACM Member

6. Honors and awards

NSF Career Award
Best Student Paper Award, Usenix ATC 2016
IETF/IRTF Applied Networking Research Prize, 2015
Best Student Paper Award, SIGCOMM 2015
Best Paper Award, NSDI 2015
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: NSDI 2019, SigComm 2018, NSDI 2018, SOSP 2017, SigComm 2017, 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.
Undergraduate Coordinator, 2018
Chair of Graduate Admissions Committee, 2016
Graduate Admissions Committee, 2013, 2014
Undergraduate Scholarships Committee, 2012, 2017

Undergraduate Admissions Committee, 2012

Coordinator of Undergraduate Direct Admits Seminar, 2011, 2012, 2017

8. Briefly list the most important publications and presentations from the past five years

Slim: OS Kernel Support for a Low-Overhead Container Overlay Network

D. Zhuo, K. Zhang, Y. Zhu, H. Liu, M. Rockett, A. Krishnamurthy, T. Anderson, *NSDI*, 2019.

Stable and Practical AS Relationship Inference with ProbLink

Y. Jin, C. Scott, A. Dhamdhere, V. Giotsas, A. Krishnamurthy, S. Shenker, *NSDI*, 2019.

TVM: An Automated End-to-End Optimizing Compiler for Deep Learning

T. Chen, T. Moreau, Z. Jiang, L. Zheng, E. Yan, H. Shen, M. Cowan, L. Wang, Y. Hu, L. Ceze, C.

Guestrin, A. Krishnamurthy, *OSDI*, 2018.

Revisiting Network Support for RDMA

R. Mittal, A. Shpiner, A. Panda, E. Zahavi, A. Krishnamurthy, S. Ratnasamy, S.

Shenker, *SIGCOMM*, 2018.

Approximating Fair Queueing on Reconfigurable Switches

N. Sharma, M. Liu, K. Atreya, A. Krishnamurthy, *NSDI*, 2018.

IncBricks: Toward In-Network Computation with an In-Network Cache

M. Liu, L. Luo, J. Nelson, L. Ceze, A. Krishnamurthy, K. Atreya, *ASPLOS*, 2017.

9. Briefly list the most recent professional development activities

None.

Kemelmacher-Shlizerman, Ira (Irena)

1. Education – degree, discipline, institution, year

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 – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

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 – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

Consultant, Google, March-Oct 2010

4. Certifications or professional registrations

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. 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

[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

Kohno, Tadayoshi

1. PhD, Computer Science, UC San Diego, 2006
MS, Computer Science, UC San Diego, 2004
BS, Computer Science, University of Colorado, 1999
2. University of Washington, Professor, 2016-present, full time
University of Washington, Associate Professor, 2011-present, full time (except sabbatical in 2014)
University of Washington, Assistant Professor, 2006-2011, full time
3. Microsoft Research, 2014, full time (sabbatical)
Cigital, Cryptographer, cryptography and computer security consultant, 2000-2001, full time
Counterpane Systems, Cryptographer, cryptography and computer security consultant, 1999-2000
4. No certifications or professional registrations
5. Senior Member, Institute of Electrical and Electronics Engineers (IEEE)
Member, Association for Computing Machinery (ACM)
6. 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
Test of Time Award at IEEE S&P (2019)
Best paper awards at IEEE S&P (2012, 2008), EuroSys (2011), CPDP (2011), USENIX Security (2009), MobiSys (2008), CCS (2002), ACSAC (2000)
7. Conference and workshop chairmanships (WSSP 2018, WWW S&P Track 2017, UPSIDE 2014, 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 2010-15)
UW CSE Department Executive Committee (2010-2011) and Faculty Recruiting Committee (2010-2013, 2014-2016, 2017-2019 (Chair))
Associate Director, Technology Policy Lab, 2013-present
8. 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.
Peter Ney, Karl Koscher, Lee Organick, Luis Ceze, and Tadayoshi Kohno. Computer security and privacy in DNA sequencing. In USENIX Security Symposium, August 2017.
9. Invited participant, National Effective Teaching Institute (NETI), American Society for Engineering Education, 2013
Regular meetings with the UW Center for Engineering Learning & Teaching

Lin, Huijia (Rachel)

1. Education –
PhD Computer Science, Cornell University, 2012
B.S. Computer Science, Zhejiang University, 2004
2. Academic experience –
Associate Professor, Computer Science & Engineering, U. Washington, 2019-present.
Assistant Professor, University of California, Santa Barbara, 2014-2018
Postdoctoral Associate, CSAIL, MIT, 2011-2013.
3. Non-academic experience -
Intern, IBM Research, 2006 & 2010
Intern, Microsoft Research, 2009
Intern, Google, 2007
4. Certifications or professional registrations
5. Current membership: IACR
6. Honors and awards:
Best Paper Award, Eurocrypt 2018
Best Paper Award, Honorable Mention, Eurocrypt 2016
NSF CAREER Award, 2017
Hellman Fellowship, 2017
Microsoft Research PhD Fellowship, 2009
7. Selected service activities: National and international:
Steering Committee Member, “Theory of Cryptography Conference” 2018–present.
Program committee member for Crypto, Eurocrypt, TCC, FOCS, STOC.
8. Selected Publications:

Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai.
Indistinguishability Obfuscation Without Multilinear Maps: New Paradigms via Low Degree Weak Pseudorandomness and Security Amplification.
To appear: The 39th International Cryptology Conference (CRYPTO 2019)

Fabrice Benhamouda, Huijia Lin.
k-Round MPC from k-Round OT via Garbled Interactive Circuits.
The 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2018)

Huijia Lin, Rafael Pass, Pratik Soni.
Two-Round and Non-interactive Concurrent Non-Malleable Commitments from Time-Lock Puzzles.
The 57th IEEE Symposium on Foundations of Computer Science (FOCS 2017)

Huijia Lin and Stefano Tessaro.
Indistinguishability Obfuscation from Trilinear Maps and Block-Wise Local PRGs.
The 37th International Cryptology Conference (CRYPTO 2017)

Huijia Lin.

Indistinguishability Obfuscation from SXDH on 5-linear Maps and Locality-5 PRGs.
The 37th International Cryptology Conference (CRYPTO 2017)

Huijia Lin

Indistinguishability Obfuscation from Constant-degree Graded Encodings.
The 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques
(EUROCRYPT 2016).

Lee, James R.

1. Education –
PhD Computer Science, University of California, Berkeley, 2005
Computer Science & Mathematics, Purdue University, 2001
B.S.
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.

- Levy, Henry M.

1. Education – degree, discipline, institution, year.
MS, Computer Science, University of Washington, 1981
BS, Math/Computer Science, Carnegie Mellon University, 1974
2. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time.
Current Position: Director and Wissner-Slivka Chair in Computer Science & Engineering, Paul G. Allen School of Computer Science & 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, Allen School Founding Director 2017). Research in operating systems, computer architecture, multithreaded computer systems, object-oriented systems and languages, performance evaluation, computer security, distributed systems, the Internet.
3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time
Consulting Engineer, Digital Equipment Corporation, 1974 – 1983
Co-Founder, Skytap Inc. (2006), Performant Inc. (2001)
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);
Fellow of the Institute for Electrical and Electronics Engineers (2002);
Microsoft Endowed Professorship in Computer Science and Engineering (1999);
Fellow of the Association for Computing Machinery (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. 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

Diamond: Automating Data Management and Storage for Wide-Area Reactive Applications, I. Zhang, N. Lebeck, P. Fonseca, B. Holt, R. Cheng, A. Norberg, A. Krishnamurthy. USENIX Conf. on Operating Systems Design and Implementation, 2016.

Customizable and Extensible Deployment of Mobile/Cloud Applications, I. Zhang, A. Szekeres, D. Van Aken, I. Ackerman, S.D. Gribble, A. Krishnamurthy. USENIX Conf. on Operating Systems Design and Implementation, 2014.

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).

9. Briefly list the most recent professional development activities: None

Mones, Barbara

1. Post Grad Animation Cert, Sheridan College, Ontario Canada. Rhode Island School of Design, MFA, University of Michigan, BFACreative **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, 3**Associate Professor and Founding Director**, Visual Information Technologies MA/MFA Graduate Program George Mason University, Fairfax, Virginia 1992-1999
2. 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
3. SIGGRAPH Member
4. SIGGRAPH, ASIFA
5. NASA Group Achievement Award, GLOBE Project, Film Screening awardsNominations 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.
8. 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
9. Research projects include “The Cinematic Sandbox: Fast Prototyping for DigitalAnimation” 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.
10. 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.

Oskin, Mark

1. 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. 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. 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. ACM Member
6. 2001 NSF CAREER Award
2005 Sloan Research Fellowship
7. 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. Martha Mergaldi-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)

Patel, Shwetak N.

Computer Science & Engineering, Electrical & Computer Engineering
shwetak@cs.washington.edu
<http://www.shwetak.com>

EDUCATION

Georgia Institute of Technology Ph.D., Computer Science	8/2003 – 8/2008
--	-----------------

Georgia Institute of Technology B.S., Computer Science	8/2000 – 5/2003
---	-----------------

Academic Experience Washington Research Foundation Endowed Professor (Full Professor), University of Washington	9/2014 - Present
---	------------------

Associate Professor, University of Washington	6/2013 – 9/2014
---	-----------------

Assistant Professor, University of Washington	9/2008 – 6/2013
---	-----------------

Non-Academic Experience Google, Director	6/2017 – Present
---	------------------

Chief Scientist, Belkin International, Inc.	1/2010 – 6/2017
---	-----------------

Microsoft Research, Visiting Researcher	9/2009 – 9/2015
---	-----------------

CURRENT MEMBERSHIP IN PROFESSIONAL ORGANIZATIONS

IEEE
ACM (ACM Fellow)

SELECT HONORS AND AWARDS

ACM Prize in Computing (2019)
ACM Fellow (2016)
Presidential Early Career Awards for Scientists and Engineers (PECASE) Award (2016)
National Academy of Engineering (NAE) Gilbreth Award (2016)
World Economic Forum Young Global Scientist Award (2013)
NSF Career Award (2013)
Sloan Fellowship (2012)
MacArthur “Genius” Fellowship (2011)
Microsoft Research Faculty Fellowship (2011)
MIT TR-35 Award: MIT Technology Review’s Top Young Innovator under the age of 35 (2009)

SERVICE ACTIVITIES

Editorial Roles and Boards
IMWUT/UbiComp Associate Editor (2016-current)
Georgia Tech GVU Advisory Board (2014-current)

IEEE Pervasive Computing Editorial Board (2014-current)

University and Department Service

Associate Director of Research and Innovation for the Allen School of Computer Science & Engineering (2016 – current)

UW Global Innovation Exchange (GIX) Director and CTO (2015-current)

Federal and State Government

Computing Community Consortium (CCC) Council Member (2015-current)

MOST IMPORTANT PUBLICATIONS (past 5 years)

Gulati, M., Parizi, F., Whitmire, E., Gupta, S., Ram, S., Singh, A., Patel, S.N.: CapHarvester: A Stick-on Capacitive Energy Harvester Using Stray Electric Field from AC Power Lines. IMWUT 2(3): 110:1-110:20 (UbiComp 2018). 2018.

Wang, E.J., Zhu, J., Jain, M., Lee, T., Saba, E., Nachman, L., Patel, S.N.: Seismo: Blood Pressure Monitoring using Built-in Smartphone Accelerometer and Camera. CHI 2018: 425.

Mariakakis, A., Banks, M., Phillipi, L., Yu, L., Taylor, J., Patel, S.N.: BiliScreen: Smartphone-Based Scleral Jaundice Monitoring for Liver and Pancreatic Disorders. IMWUT 1(2): 20:1-20:26. (UbiComp 2017). 2017.

Wang, E.J., Li, W., Hawkins, D., Grenheimer, T., Norby-Slycord, C., Patel, S.N., HemaApp: Noninvasive Blood Screening of Hemoglobin using Smartphone Cameras. Proceedings of the ACM International Conference on Ubiquitous Computing (UbiComp 2016). pp. 593-604. 2016. (Acceptance Rate: 24%)

Goel, M., Saba, E., Stiber, M., Whitmire, E., Fromm, J., Larson, E.C., Borriello, G., Patel, S.N., SpiroCall: Measuring Lung Function over a Phone Call. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2016). pp. 5675-5685. 2016. (Acceptance Rate: 23%)

Ravichandran, R., Saba, E., Chen, K., Goel, M., Gupta, S., Patel, S.N. WiBreathe: Estimating respiration rate using wireless signals in natural settings in the home. In the Proceedings of the IEEE International Conference on Pervasive Computing and Communications (Percom 2015). pp. 131-139. 2015. (Acceptance Rate: 8%)

Zhao, C., Chen, K., Aumi, Md. T., Patel, S.N., Reynolds, M.S. SideSwipe: detecting in-air gestures around mobile devices using actual GSM signal. In the Proceedings of the ACM Symposium on User Interface Software and Technology UIST 2014. pp 527-534. 2014. (Acceptance Rate: 22%)

Greef, L., Goel, M., Seo, M.J., Larson, E.C., Stout, J.W., Taylor, J.A., Patel, S.N. Bilicam: using mobile phones to monitor newborn jaundice. In the Proceedings of UbiComp 2014. pp.331-342. 2014. (Acceptance Rate: 21%)

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%)

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%)

Perkins, Hal

1. Education
M.S., Computer Science, Cornell University, 1982
B.S., Mathematics, Arizona State University, 1975.
2. 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. 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. Member, ACM; Member, IEEE
6. 2011-12 UW ACM Teaching Award
7. College of Engineering Council on Educational Policy, 2010-2016
College of Engineering CEP Subcommittee on Admissions & Enrollments, 2018-2019
UW CSE ACM chapter faculty advisor, 2010-2013
UW CSE 5th year Master's admissions committee, 2014-2015
UW CSE PMP admissions committee, 2004-2012, 2014-2016
UW CSE Undergraduate admissions committee, 2016-2018
UW CSE Undergraduate curriculum committee, 2018-2019
UW CSE Non-Majors curriculum committee, 2011-12
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: n/a
9. Briefly list the most recent professional development activities: n/a

Rao , Anup

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 –
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:
Conference Committee: STOC 2009, CCC 2011, STOC 2015, RANDOM 2015 (Chair), CCC 2016, ITCS 2016
8. Selected Publications:

Sivaramakrishnan Natarajan Ramamoorthy, Anup Rao: Non-Adaptive Data structure Lower Bounds for Median and Predecessor Search from Sunflowers. IEEE Conference on Computation Complexity, 2018.

Markrand Sinha, Anup Rao: Simplified Separation of Information and Communication. Theory of Computing, 2017.

Christian Engels, Mohit Garg, Kazuhisa Makino, Anup Rao: On Expressing Majority as a Majority of Majorities. Electronic Colloquium on Computational Complexity (ECCC) 24: 174 (2017).

Reges, Stuart

1. Education –

9/79-6/82 **Stanford University**, MS in Computer Science

8/77-5/79 **Case Western Reserve University**, BS in Mathematics

2. Academic Experience -

9/08-present **Principal Lecturer**, Computer Science & Eng., University of Washington. 7/04-8/08

Senior Lecturer, Computer Science & Eng., University of Washington

5/01-6/04 **Associate Head for Undergraduate Studies**, Department of Computer Science, University of Arizona.

8/96-6/04 **Senior Lecturer**, Department of Computer Science, University of Arizona.

12/93-9/01 **Management/Programming Consultant**

9/88-5/91 **Senior Lecturer**, Department of Computer Science, Stanford University.

9/85-3/88 **Assistant Chairman for Education**, Department of Computer Science, Stanford University.

Awards and Activities

- Elected as Faculty Representative, Computer Science Teachers Association (2012-2013)
- Distinguished Teaching Award (UW, 2011)
- Student ACM Teaching Award (UW, 2011)
- Second Place, Student ACM Teaching Award (UW, 2007)
- Honorable Mention, Student ACM Teaching Award (UW, 2006)
- Honors College Outstanding Advisor (U of A, 2002)
- College of Science Distinguished Teaching Award (U of A, 1998)
- Honorable Mention, Associated Students of Stanford University Teaching Award (Stanford, 1991)
- School of Engineering Distinguished Advisor Award (Stanford, 1986)
- Dinkelspiel Award for Outstanding Service to Undergraduate Education (Stanford, 1985)

Textbooks

- *Building Java Programs*, 2007, Addison Wesley, coauthored with Marty Stepp
- *Pascal and Beyond: Data Abstraction and Data Structures Using Turbo Pascal*, co-authored with Steve Fisher, 1992, Wiley & Sons
- *Building Pascal Programs*, 1987, Little, Brown

Refereed Papers

- “Broadening Participation: The Why and the How,” with Crystal Eney, Ed Lazowska, and Hélène Martin, IEEE Computer (2013)
- “The Mystery of $b := (b == \text{false})$,” Thirty-ninth SIGCSE Technical Symposium on Computer Science Education (2008), included by request in Annals of Research on Engineering Education (2009)
- “Back to Basics in CS1 and CS2,” Thirty-seventh SIGCSE Technical Symposium on Computer Science Education (2006)
- “Using Undergraduates as Teaching Assistants at a State University,” Thirty-fourth SIGCSE Technical Symposium on Computer Science Education (2003)

- “Can C# Replace Java in CS1 and CS2?,” Seventh ITICSE Conference (2002)
- “Conservatively Radical Java in CS1,” Thirty-first SIGCSE Technical Symposium on Computer Science Education (2000)
- “The Effective Use of Undergraduates to Staff Large Introductory Courses,” Nineteenth SIGCSE Technical Symposium on Computer Science Education (1988)

Workshops and Other Activities

- Invited to represent a “sage elder” in the “New Teaching Faculty Roundtable,” SIGCSE Technical Symposium on Computer Science Education (2011)
- Invited to represent a “sage elder” in the “New Teaching Faculty Roundtable,” SIGCSE Technical Symposium on Computer Science Education (2010)
- Associate Program Chair, SIGCSE Technical Symposium on Computer Science Education (2010)
- Invited to represent a “sage elder” in the “New Teaching Faculty Roundtable,” SIGCSE Technical Symposium on Computer Science Education (2009)
- “What’s New in CS1 and CS2,” panel presentation at the Northwest Regional Conference of the Consortium for Computing Sciences in Colleges (2005)
- Reader for Advanced Placement Computer Science: 2001, 2006, 2007, 2011
- College Board workshop consultant since 2001: presenting one-day workshops to high school teachers about the Advanced Placement Exam in Computer Science
- “Teaching C# in CS1, CS2, and Advanced Programming Courses,” workshop at the Thirty-fourth SIGCSE Technical Symposium on Computer Science Education (2003)
- “Teaching Programming with .Net,” 3rd Annual .Net Technologies in Curriculum Workshop (2003)
- “Experiences teaching C#,” 2nd Annual .Net Technologies in Curriculum Workshop (2002)
- “Plans for a course in C# for Java Programmers,” 1st Annual .Net Technologies in Curriculum Workshop (2001)

Reinecke, Katharina

Education:

Ph.D. Computer Science, University of Zurich, 2010

Diploma/M.S. Computer Science, University of Koblenz, 2006

Academic Experience:

Assistant Professor, Computer Science & Engineering, U. Washington, 2015-Present

Assistant Professor, School of Information, U. Michigan, 2014-2015

Postdoctoral Fellow, Computer Science, Harvard University, 2010-2013

Non-Academic Experience:

Certifications or Professional Registrations:

Current Membership:

Association for Computing Machinery (ACM)

Honors and Awards:

Madrona Prize, 2017

National Science Foundation CAREER Award, 2016

Best Paper Award, CHI 2017

Best Paper Award, CHI 2016

Best Paper Honorable Mention, CSCW 2015

European Research Paper of the Year Award, 2014

Best Paper Honorable Mention, CHI 2013

Best Paper Award, CHI 2013

Mercator Prize, 2011

Best Paper Award, UMAP 2009

Swiss Computer Science Challenge Award, 2008

Best Student Award, 2007

Selected Service Activities:

National and International:

Conference Program Co-Chair, UIST 2019

Program Committee Learning at Scale, 2018

Program Committee Human Factors in Computing Systems (CHI), 2014, 2015, 2017, 2018

Associate Editor International Journal of Human-Computer Studies (IJHCS), 2015-2017

Program Committee Computer-Supported Cooperative Work (CSCW), 2017

AAAI Conference on Human Computation and Crowdsourcing (HCOMP) Works-in-Progress & Demonstration Co-Chair, 2017

Program Committee AAAI Conference on Human Computation and Crowdsourcing (HCOMP), 2016

Program Committee International Workshop on Diversity-Aware Artificial Intelligence @ ECAI, 2016

Program Committee Human Factors in Computing Systems (CHI) Workshops, 2015

Program Committee User Modeling, Adaptation, and Personalization (UMAP), 2010, 2014, 2015

User Modeling, Adaptation, and Personalization (UMAP) late-breaking results program committee, 2015

User Modeling, Adaptation, and Personalization (UMAP) posters and demos program committee, 2015

Program Committee Workshop on Personalised Multilingual Web Access at World Wide Web (WWW), 2014, 2015

Program Committee Designing Interactive Systems (DIS), 2014

Program Committee 29th ACM Symposium On Applied Computing (SAC), 2014

Program Committee Workshop on Culturally-Aware Tutoring Systems at Artificial Intelligence in Education (AIED), 2013

Program Committee Intelligent User Interfaces (IUI), 2011

Program Committee Doctoral Consortium at User Modeling, Adaptation, and Personalization (UMAP), 2010

University of Washington Service:

CSE Diversity Committee (January 2017-present)

BS/MS Committee (2018-2019)

CSE Graduate Admissions Committee (2017-2018)

MHCI+D Executive Committee (2016-2018)

Co-chair of the University of Washington DUB 2016 Annual Retreat

Selected Publications:

Roesner, Franziska

1. Education:
 - University of Washington, Computer Science and Engineering, Ph.D. 2014
 - University of Washington, Computer Science and Engineering, M.S. 2011
 - The University of Texas at Austin, Computer Science, B.S. 2008
 - The University of Texas at Austin, Plan II Honors (Liberal Arts), B.A. 2008
2. Academic experience:
 - University of Washington:
 - Paul G. Allen School of Computer Science and Engineering, Assistant Professor (2014–Present)
 - Electrical Engineering Department, Adjunct Assistant Professor (2016–Present)
 - Department of Human Centered Design and Engineering, Adjunct Assistant Professor (2017–Present)
3. Non-academic experience:
 - Internships
 - Google – Software Engineering, Summer 2012
 - Microsoft Research – Research, Summer 2010
 - Amazon – Security Engineering, Summer 2009
 - Amazon – Software Engineering, Summer 2008
4. Certifications or professional registrations:
5. Current membership in professional organizations:
 - Association for Computing Machinery (ACM)
 - The Advanced Computing Systems Association (USENIX)
 - Institute of Electrical and Electronics Engineers (IEEE)
6. Honors and awards (selected):
 - MIT Technology Review Innovator Under 35 (2017)
 - Emerging Leader Award, Hall of Honor, UT Austin College of Natural Sciences (2017)
 - Google Security and Privacy Research Award (2017)
 - NSF CAREER Award (2017)
 - William Chan Memorial Dissertation Award (2014)
 - IEEE Security & Privacy Best Practical Paper Award (2012)
7. Service activities (selected):
 - Program Co-Chair, USENIX Security 2020
 - Program Co-Chair, USENIX Enigma, 2018 and 2019
 - Member, DARPA Information Science and Technology (ISAT) study group, 2016–Present
 - Program Co-Chair, USENIX Summit on Hot Topics in Security (HotSec), 2016
 - Program Committee Member (Selected): USENIX Security Symposium (2015, 2016, 2017, 2019), IEEE Symposium on Security & Privacy (2015, 2016, 2017, 2019), Privacy Enhancing Technologies Symposium PETS (2018, 2019), etc.
 - Allen School Undergraduate Admissions Committee (2018, 2019)
8. Most important publications and presentations from the past five years:

- E. Zeng and F. Roesner. Understanding and Improving Security and Privacy in Multi-User Smart Homes: A Design Exploration and In-Home User Study. 28th USENIX Security Symposium, August 2019.
 - K. Ruth, T. Kohno, and F. Roesner. Secure Multi-User Content Sharing for Augmented Reality Applications. 28th USENIX Security Symposium, August 2019.
 - C. Chen, N. Dell, and F. Roesner. Computer Security and Privacy in the Interactions Between Victim Service Providers and Human Trafficking Survivors. 28th USENIX Security Symposium, August 2019. ^[1]_[SEP]
 - C. Geeng and F. Roesner. Who's In Control?: Interactions In Multi-User Smart Homes. ACM ^[1]_[SEP]Conference on Human Factors in Computing Systems (CHI), May 2019. ^[1]_[SEP]
 - K. Lebeck, T. Kohno, and F. Roesner. Enabling Multiple Applications to Simultaneously Augment ^[1]_[SEP]Reality: Challenges and Directions. 20th Workshop on Mobile Computing Systems and Applications ^[1]_[SEP](HotMobile), February 2019. ^[1]_[SEP]
 - J. Akers, G. Bansal, G. Cadamuro, C. Chen, Q. Chen, L. Lin, P. Mulcaire, R. Nandakumar, M. ^[1]_[SEP]Rockett, L. Simko, J. Toman, T. Wu, E. Zeng, B. Zorn, and F. Roesner. Technology-Enabled Disinformation: Summary, Lessons, and Recommendations. University of Washington Technical Report UW-CSE-18-12-02 and arXiv:1812.09383, December 2018. Outcome of my graduate special-topics course on technology-enabled disinformation (CSE 599B, Fall 2018). ^[1]_[SEP]
 - K. Lebeck, K. Ruth, T. Kohno, and F. Roesner. Towards Security and Privacy for Multi-User Augmented Reality: Foundations with End Users. 39th IEEE Symposium on Security and Privacy (Oakland), May 2018. ^[1]_[SEP]
 - L. Simko, S. Ibtasam, A. Lerner, F. Roesner, and T. Kohno. Computer Security and Privacy for Refugees in the United States. 39th IEEE Symposium on Security and Privacy (Oakland), May 2018. ^[1]_[SEP]
 - Lerner, T. Kohno, and F. Roesner. Rewriting History: Changing the Archived Web from the Present. 24th ACM Conference on Computer and Communications Security (CCS), November 2017. ^[1]_[SEP]
 - S. E. McGregor, E. Watkins, M. N. Al-Ameen, K. Caine, and F. Roesner. When the Weakest Link is Strong: Secure Collaboration in the Case of the Panama Papers. 26th USENIX Security Symposium, August 2017. ^[1]_[SEP]
 - E. Zeng, S. Mare, and F. Roesner. End User Security and Privacy Concerns with Smart Homes. 13th Symposium on Usable Privacy and Security (SOUPS), July 2017. ^[1]_[SEP]
 - K. Lebeck, K. Ruth, T. Kohno, and F. Roesner. Securing Augmented Reality Output. 38th IEEE Symposium on Security and Privacy (Oakland), May 2017. Selected for invitation to special issue of IEEE Security & Privacy Magazine. ^[1]_[SEP]
 - E. McReynolds, S. Hubbard, T. Lau, A. Saraf, M. Cakmak, and F. Roesner. Toys That Listen: A Study of Parents, Children, and Internet-Connected Toys. ACM Conference on Human Factors in Computing Systems (CHI), May 2017. ^[1]_[SEP]
 - Lerner, E. Zeng, and F. Roesner. Confidante: Usable Encrypted Email - A Case Study With Lawyers and Journalists. 2nd IEEE European Symposium on Security and Privacy (EuroS&P), April 2017. ^[1]_[SEP]
 - T. Ringer, D. Grossman, and F. Roesner. AUDACIOUS: User-Driven Access Control with Unmodified Operating Systems. 23rd ACM Conf. on Computer and Communications Security (CCS), Oct. 2016. ^[1]_[SEP]
 - Lerner, A. Kornfeld Simpson, T. Kohno, and F. Roesner. Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016. 25th USENIX Security Symposium, August 2016. ^[1]_[SEP]
 - K. Lebeck, T. Kohno, and F. Roesner. How to Safely Augment Reality: Challenges and Directions. 17th Workshop on Mobile Computing Systems & Applications (HotMobile), Feb. 2016. ^[1]_[SEP]
 - S. E. McGregor, P. Charters, T. Holliday, and F. Roesner. Investigating the Computer Security Practices and Needs of Journalists. 24th USENIX Security Symposium, August 2015. ^[1]_[SEP]
9. Briefly list the most recent professional development activities:

Ruzzo, Walter L.

1. 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
5. Current membership in professional organizations
6. Honors and awards
7. Service activities (within and outside of the institution)
8. Publications

Books

- Gorodkin, Ruzzo (eds.) *RNA Sequence, Structure and Function: Computational and Bioinformatic Methods*. Methods in Mol Bio. Humana Press, 2014. (Springer Protocols).

Refereed Journals

- Torarinsson, Yao, Wiklund, Bramsen, Hansen, Kjems, Tommerup, **Ruzzo**, Gorodkin. “Comparative genomics beyond sequence-based alignments: RNA structures in the ENCODE regions.” *Genome Res*. 2008 Feb;18(2):242-51. [PMID: 18096747](#)
- Weinberg, Regulski, Hammond, Barrick, Yao, **Ruzzo**, Breaker. “The aptamer core of SAM-IV riboswitches mimics the ligand-binding site of SAM-I riboswitches.” *RNA*. Mar, 2008, 14(5):822-828; [PMID: 18369181](#)
- Regulski, Moy, Weinberg, Barrick, Yao, **Ruzzo**, Breaker. “A widespread riboswitch candidate that controls bacterial genes involved in molybdenum cofactor and tungsten cofactor metabolism.” *Mol Microbiol*. May, 2008, 68(4):918-32 [PMID: 18363797](#) [PMC2408646](#)
- 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**. “MultiPerm: 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.
- Z. Yao, K.L. Macquarrie, A.P. Fong, S.J. Tapscott, **W.L. Ruzzo**, and R.C. Gentleman. Discriminative motif analysis of high-throughput dataset. *Bioinformatics*, 30(6):775–783, Mar 2014. PMID:24162561.
- X. Peng, L. Pipes, H. Xiong, R.R. Green, D.C. Jones, **W.L. Ruzzo**, G.P. Schroth, C.E. Mason, R.E. Palermo, and M.G. Katze. Assessment and improvement of Indian-origin rhesus macaque and Mauritian-origin cynomolgus macaque genome annotations using deep transcriptome sequencing data. *J. Med. Primatol.*, 43(5):317–328, Oct 2014. PMID:24810475.
- K.T. Kuppusamy, D.C. Jones, H. Sperber, A. Madan, K.A. Fischer, M.L. Rodriguez, L. Pabon, W.Z. Zhu, N.L. Tulloch, X. Yang, N.J. Sniadecki, M.A. Laflamme, **W.L. Ruzzo**, C.E. Murry, and H. Ruohola-Baker. Let-7 family of microRNA is required for maturation and adult-like metabolism in stem cell-derived cardiomyocytes. *Proc. Natl. Acad. Sci. U.S.A.*, 112(21):E2785–2794, May 2015. PMID:25964336.
- N.J. Palpant, L. Pabon, M. Roberts, B. Hadland, D. Jones, C. Jones, R.T. Moon, **W.L. Ruzzo**, I. Bernstein, Y. Zheng, and C.E. Murry. Inhibition of β -catenin signaling respecifies anterior-like endothelium into beating human cardiomyocytes. *Development*, 142(18):3198–3209, Sep 2015. PMID:25964336.
- S.E. Seemann, A.H. Mirza, C. Hansen, C.H. Bang-Berthelsen, C. Garde, M. Christensen-Dalsgaard, E. Torarinsson, Z. Yao, C.T. Workman, F. Pociot, H. Nielsen, N. Tommerup, **W.L. Ruzzo**, and J. Gorodkin. The identification and functional annotation of RNA structures conserved in vertebrates. *Genome Research*, 27(8):1371–1383, Aug 2017. PMID:28487280.
- N. Hecker, S.E. Seemann, A. Silahatoglu, **W.L. Ruzzo**, and J. Gorodkin. Associating transcription factors and conserved RNA structures with gene regulation in the human brain. *Scientific Reports*, 7(1):5776, Jul 2017. PMID:28720872.
- N. J. Palpant, Y. Wang, B. Hadland, R.J. Zaunbrecher, M. Redd, D. Jones, L. Pabon, R. Jain, J. Epstein, **W.L. Ruzzo**, Y. Zheng, I. Bernstein, A. Margolin, and C.E. Murry. Chromatin and transcriptional analysis of mesoderm progenitor cells identifies HOPX as a regulator of primitive hematopoiesis. *Cell Rep*, 20(7):1597–1608, Aug 2017. PMID:28813672.
- J.A. Koester, C.T. Berthiaume, N. Hiranuma, M.S. Parker, V. Iverson, R. Morales, **W.L. Ruzzo**, and E.V. Armbrust. Sexual ancestors generated an obligate asexual and globally dispersed clone within the model diatom species *Thalassiosira pseudonana*. *Scientific Reports*, 8(1):10492, July 2018. PMID:30002405.
- R. Kirsch, S.E. Seemann, **W.L. Ruzzo**, S.M. Cohen, P.F. Stadler, and J. Gorodkin. Identification and characterization of novel conserved RNA structures in *Drosophila*. *BMC Genomics*, 19(1):899, Dec. 2018.

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.)**

- Brossard, Richmond, Green, Ebeling, **Ruzzo**, Olson, Hauck. A data-intensive programming model for FPGAs: A genomics case study. Symp Appl Accelerators in High-Performance Comp(SAAHPC'12), Lemont, IL, July 2012.
 - McVicar, **Ruzzo**, Hauck. Accelerating ncRNA homology search with FPGAs. 21st ACM/SIGDA Intl Symp on Field-Programmable Gate Arrays (FPGA 2013), Monterey, CA, Feb. 2013.
9. Briefly list the most recent professional development activities

Schafer, Hunter

1. Education – degree, discipline, institution, year
 - M.S. Computer Science & Engineering, University of Washington, 2018
 - B.S. Computer Science & Engineering, University of Washington, 2016
2. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time
 - University of Washington, Lecturer, 2018 – present, full time
 - University of Washington, Predoctoral Instructor, Summer 2017, full time
3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time
 - Sift Science, Software Developer, Summer 2019, full time
 - Socrata, Software Developer, Summer 2018, full time
 - Redfin, Software Developer, Summer 2016, full time
4. Certifications or professional registrations
 - None
5. Current membership in professional organizations
 - ACM, ACM-SIGCSE
6. Honors and awards
 - None
7. Service activities (within and outside of the institution)
 - None
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
 - None
9. Briefly list the most recent professional development activities
 - ACM Special Interest Group on Computer Science Education (SIGCSE) Technical Symposium, 2017 - present
 - Consortium for the Advancement of Undergraduate STEM Education (CAUSE) Cohort Member, 2018 - present

Seelig, Georg

1. Education

- Ph.D., Physics, University of Geneva, Switzerland, 2003
- Diploma, Physics, University of Basel, 1999

9. Academic experience

- Associate Professor of Electrical Engineering, University of Washington, dates
- Since 9/2015: Associate professor, Electrical & Computer Engineering and Paul G. Allen School for Computer Science & Engineering, University of Washington, Seattle
- 3/2009 - 9/2015: Assistant professor, Electrical Engineering and Computer Science & Engineering, University of Washington, Seattle
- 10/2003-3/2009: Postdoctoral researcher, California Institute of Technology
- 10/1999-10/2003: Graduate research assistant, University of Geneva, Switzerland

Honors and awards

- MSR Outstanding Collaborator Award, 2016, Microsoft Research
- Outstanding Faculty Award, 2015, Department of Electrical Engineering
- Young Investigator Award, 2014, ONR
- Young Faculty Award, 2012, DARPA
- Sloan Research Fellowship, 2011, Alfred P. Sloan Foundation
- CAREER Award, 2010, National Science Foundation
- Career Award at the Scientific Interface, 2008, Burroughs Wellcome Fund
- Fellowship for Advanced Researchers, 2006, Swiss National Science Foundation
- Postdoctoral Fellowship, 2004, Center for Biological Circuit Design, Caltech
- Postdoctoral Fellowship, 2003, Swiss National Science Foundation

Service activities (within and outside of the institution)

- Departmental Service: Various committees in ECE and CSE including multiple faculty search committees, admissions committees and executive committee
- Participation in CoE BRIDGE program, 2009, 2010, 2011, 2014, 2017
- Peer review: Cell, Science, Nature, Nature Chemistry, Nature Nanotechnology, Nature Biotechnology, Nature Communications, Nature Chemical Biology, Nature Methods, Nature Materials, Genome Research, PNAS, Molecular Systems Biology, and more.
- Proposal reviewing: NIH, NSF, Swiss National Science Foundation, Netherlands Organisation for Scientific Research and others
- Conference organization: DNA 25 (2019), SyntheticBiology@UW (2013), BASC: Brain, Analysis, Synthesis, Computation workshop for ISAT (2014)

Most important publications and presentations from the past five years

- Alexander B. Rosenberg*, Charles Roco*, Richard A. Muscat, Anna Kuchina, Paul Sample, Sumit Mukherjee, Wei Chen, David J Peeler, Zizhen Yao, Suzie H Pun, Drew L. Sellers, Bosiljka Tasic, and Georg Seelig, *Single cell profiling of the developing mouse brain and spinal cord with split-pool barcoding*, Science **360**, 176 (2018).
- Randolph Lopez, Ruofan Wang, and Georg Seelig, *A molecular multi-gene classifier for disease diagnostics*, Nature Chemistry **10**, 746 (2018).
- Gourab Chatterjee, Neil Dalchau, Richard A Muscat¹, Andrew Phillips and Georg Seelig, *A spatially localized architecture for fast and modular DNA computing*, Nature Nanotechnology **12**, 920 (2017).

- Josh T. Cuperus*, Benjamin Groves*, Anna Kuchina*, Alexander Rosenberg*, Nebojsa Jojic, Stanley Fields and Georg Seelig, *Deep learning of the regulatory grammar of yeast 5' untranslated regions from 500,000 random sequences*, Genome Research **27**, 2015 (2017).
- Alexander B. Rosenberg², Rupali Pathwardhan, Jay Shendure, and Georg Seelig, *Learning the Sequence Determinants of Alternative Splicing from Millions of Random Sequences*, Cell **163**, 698 (2015).

Seitz, Steve

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. Current membership in professional organizations: IEEE, ACM
5. 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
6. 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

7. 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

Shapiro, Linda G.

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:

E. Mercan, S. Aksoy, L. G. Shapiro, D. L. Weaver, T. T. Brunye, J. G. Elmore, "Localization of Diagnostically Relevant Regions of Interest in Whole Slide Images: A Comparative Study," *Journal of Digital Imaging*, 2016.

J. Wu, C. Heike, C. Birgfeld, K. Evans, M. Maga, C. Morrison, B. Saltzman, L. Shapiro, R. Tse, "Measuring Symmetry in Children with Unrepaired Cleft Lip. Defining a Standard for the 3D Mid-Facial Reference Plane," *Cleft Palate Craniofacial Journal*, 2016.

J. Wu, S. Liang, L. Shapiro, R. Tse, "Measuring Symmetry in Children with Cleft Lip. Part 2: Quantification of Nasolabial Symmetry Before and After Cleft Lip Repair," *Cleft Palate Craniofacial Journal*, 2016.

S. Liang, L. Shapiro, R. Tse, "Measuring Symmetry in Children with Cleft Lip. Part 3: Quantifying Nasal Symmetry and Nasal Normalcy Before and After Unilateral Cleft Lip Repair," *Cleft Palate Craniofacial Journal*, 2016.

D. B. Nagarkar, E. Mercan, D. L. Weaver, T. T. Bruny'e, P. A. Carney, M. H. Rendi, A. H. Beck, P. D. Frederick, L. G. Shapiro, and J. G. Elmore. "Region of Interest Identification and Diagnostic Agreement in Breast Pathology." *Modern Pathology*, 2016.

Y. Chen, H.Lu, L. Shapiro, R. S. Travillian, L. Li, "An Approach to Semantic Query Expansion System Based on Hepatitis Ontology," *Journal of Biologic Research-Thessaloniki*, 23(Suppl 1):S11, 2016.

E. Mercan, C. S. Morrison, E. Stuhag, L. G. Shapiro, R. W. Tse, "Novel Computer Vision Analysis of Nasal Shape in Children with Unilateral Cleft Lip," *Journal of Cranio-Maxillofacial Surgery*, Elsevier, ScienceDirect, October 29, 2017.

E. Mercan, L. G. Shapiro, T. T. Brunye, D. L. Weaver, J. G. Elmore, "Characterizing Diagnostic Search Patterns in Digital Breast Pathology: Scanners and Drillers," *Journal of Digital Imaging*, July, 2017.

C. Mercan, S. Aksoy, E. Mercan, L. G. Shapiro, D. L. Weaver, J. G. Elmore, "Multi-instance Multi-label Learning for Multi-class Classification of Whole Slide Breast Histopathology Images," *Transactions on Medical Imaging*, Vol. 37, No. 1, 2018, pp. 316-325

9. Briefly list the most recent professional development activities: None for me; I mentor others.

Smith, Joshua R.

1. Education
BA, Computer Science, Philosophy, Williams College, 1991
MA, Physics and Theoretical Physics, University of Cambridge, 1997
MS / PhD, Media Arts & Sciences, MIT, 1995/1999
2. Academic experience – (all full time)
2011-2014 Associate Professor (Without Tenure), CSE & EE, UW
2014-2017 Associate Professor (With Tenure), CSE & EE UW
2017- Professor, CSE; Zeuschel Professor ECE, UW
3. Non-academic experience
1999-2004 Director Escher Labs, Chief Scientist, Escher Group, Cambridge MA
2004-2010 Principal Engineer, Intel Corp, Seattle, WA
2015 Co-Founder Jeeva Wireless; Co-Founder Wibotic
2016 Co-Founder Proprio
4. Certifications or professional registrations
5. Current membership in professional organizations
Senior Member, IEEE; Member, ACM
6. Honors and awards
11 Best paper awards; Zeuschel Professorship; Allen Distinguished Investigator
7. Service activities (within and outside of the institution)
Outside: 2019 IEEE RFID General Chair; Program Committees: 2019, 2017 Mobicom; 2016 SIGCOMM, 2016 UIST, 2016 Sensys, Editorial Board Member for journal Wireless Power Transfer.
Within: EE Professional Masters Program Faculty Coordinator; GIX Master of Science in Technology Innovation Interdisciplinary Faculty Group (IFG); Thrust Leader, Center for Neurotechnology (an NSF ERC); College of Engineering IP Committee
8. Publications: See <https://scholar.google.com/citations?user=LnAus20AAAAJ> for complete list
Books
 - Wirelessly Powered Sensor Networks and Computational RFID, Joshua R. Smith (Ed.), Springer SBM, February 2013.**Refereed Journal Papers**
 - Analysis, Experimental Results, and Range Adaptation of Magnetically Coupled Resonators for Wireless Power Transfer, Alanson P. Sample, David T. Meyer, Joshua R. Smith, IEEE Transactions on Industrial Electronics, Vol. 58, No. 2, pp 544-554, Feb 2011.
 - 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.**Refereed Conference Papers**
 - Ambient Backscatter: Wireless Communication out of Thin Air, Vincent Liu, Aaron Parks¹, Vamsi Talla¹, Shyam Gollakota, David Wetherall, Joshua R. Smith, SIGCOMM 2013. (Best Paper Award.)
9. Briefly list the most recent professional development activities

Suciu, Dan

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 – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time
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 – 2000 **Adjunct professor** at the Computer Science Department, UPenn
1990 - 1991 **Lecturer**, Polytechnic of Bucharest, Computer Science Department
1984 - 1990 **Assistant Professor**, Polytechnic of Bucharest. Computer Science Department
3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time
2018 – present **Consultant**, RelationalAI
2015 – 2017 **Consultant**, Logicblox
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
2019 SIGMOD Best Paper Award
2018 VLDB Best Demonstration Award
2017 SIGMOD Best Demonstration Award
2014 VLDB Ten Year Best Paper Award
2013 ICDE 2013 Influential Paper Award
2013 ICDT 2013 Best Paper Award
2012 SIGMOD'2012 Best Demonstration Award
2012 ACM PODS Alberto O. Mendelzon Test-of-Time Award
2011 Elected Fellow of the ACM, 2011
2010 ACM PODS Alberto O. Mendelzon Test-of-Time Award
2001 NSF Career Award
2001 Alfred P. Sloan Fellow, 2001-2002.

2000 ACM SIGMOD Best Paper Award

1996 Morris and Dorothy Rubino_ Award for best doctoral dissertation.

1994 Doctoral Fellowship from the Institute for Research in Cognitive Science, UPenn

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)

General Chair: PODS 2018-2020

PC Chair: ACM PODS 2007, SIGMOD 2017

PC Co-chair/Vice Chair: CDT, WWW, ICDE

Conference PC Member: PODS, SIGMOD, ICDE, ICDT, VLDB

Associate Editor: JACM, Information System

Department Service:

PMP Coordinator 2018-.

Graduate Program Coordinator 2011-2014.

Graduate Admissions Committee, chair 2006-2007.

Graduate Admissions Committee 2005-2006, 2008-2009, 2009-2010, 2017-2018.

Industrial Affiliates Committee 2000-2001, 2001-2002, 2002-2003, 2003-2004.

Undergraduate CS Admission Committee 2002.

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

Koutris, Salihoglu, Suciu: Algorithmic Aspects of Parallel Data Processing. Foundations and Trends in Databases, 2018

Li, Li, Miklau, Suciu: A theory of pricing private data. Commun. ACM, 2017,

Van den Broeck, Suciu: Query Processing on Probabilistic Data: A Survey. Foundations and Trends in Databases, 2017

Beame, Koutris, Suciu: Communication Steps for Parallel Query Processing. J. ACM, 2017

Salimi, Rodriguez, Howe, Suciu: Capuchin: Causal Database Repair for Algorithmic Fairness, SIGMOD, 2019

Chu, Murphy, Roesch, Cheung, Suciu: Axiomatic Foundations and Algorithms for Deciding Semantic Equivalences of SQL Queries. PVLDB, 2018

Abo Khamis, Ngo, Suciu: What Do Shannon- type Inequalities, Submodular Width, and Disjunctive Datalog Have to Do with One Another? PODS 2017

9. Briefly list the most recent professional development activities: None

Tanimoto, Steven L.

1. Education – degree, discipline, institution, year.
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 – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time
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 – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time
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. 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

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.

Tatlock, Zachary

1. Education

- a. PhD, Computer Science and Engineering, UC San Diego, 2014
- b. BS, Computer Science (Honors), Purdue University, 2007
- c. BS, Mathematics, Purdue University, 2007

2. Academic experience

University of Washington, Seattle, WA
Assistant Professor
Fall 2014 - Present

University of Washington, Seattle, WA
Acting Assistant Professor
Fall 2013 - Fall 2014

University of California, San Diego, La Jolla, CA
Graduate Researcher
Fall 2007 - Fall 2013

Microsoft Research India, Bangalore, India
Research Intern
Summer 2010

3. Non-academic experience

None

4. Certifications or professional registrations

None

5. Current membership in professional organizations

ACM SIGPLAN

6. Honors and awards

NSF CAREER Award, 2017
UW Distinguished Teaching Award Nomination, 2015
PLDI 2015 Distinguished Paper Award

7. Service activities (within and outside of the institution)

ASPLOS 2020 Program Committee
ICFP 2019 External Review Committee
PLDI 2019 Program Committee
CPP 2019 Program Committee
TOPLAS Referee 2018
PLDI 2018 Programming Languages Mentoring Workshop Speaker
ITP 2018 Program Committee
PNW PLSE 2018 Organizer and Program Committee Chair

POPL 2018 Program Committee
 POPL 2018 Programming Languages Mentoring Workshop Speaker PNW PLSE 2018 Organizing Committee Chair
 PNW PLSE 2018 Program Committee Chair
 Onward! 2017 Program Committee
 Splash 2017 Doctoral Symposium Program Committee
 ASPLOS 2017 External Reviewer
 POPL 2017 External Reviewer
 PLDI 2016 Program Committee
 PLDI 2016 Artifact Evaluation Committee co-Chair
 CPP 2016 Program Committee
 ASPLOS 2016 External Review Committee
 POPL 2016 External Review Committee
 POPL 2016 Student Research Competition Organizer
 PLDI 2015 Student Research Competition Committee CoqPL 2015 Program Committee
 POPL 2015 External Review Committee
 Inspirations at SPLASH 2014 Organizer and Speaker OOPSLA 2014 External Review Committee
 PLDI 2014 External Review Committee
 Compiler Construction 2009 External Reviewer

8. Publications

Teaching Rigorous Distributed Systems With Efficient Model Checking
 EuroSys 2019: Proceedings of the Fourteenth EuroSys Conference
 Ellis Michael, Doug Woos, Thomas E. Anderson, Michael D. Ernst, Zachary Tatlock

Functional Programming for Compiling and Decompiling Computer-Aided Design
 ICFP 2018: 23rd International Conference on Functional Programming
 Chandrakana Nandi, James R. Wilcox, Taylor Blau, Dan Grossman, Zachary Tatlock

Relay: A New IR for Machine Learning Frameworks
 MAPL 2018: Workshop on Machine Learning and Programming Languages
 Jared Roesch, Steven Lyubomirsky, Logan Weber, Josh Pollock, Tianqi Chen, Zachary Tatlock

Combining Tools for Optimization and Analysis of Floating-Point Computations
 FM 2018: International Symposium on Formal Methods
 Heiko Becker, Pavel Panchekha, Eva Darulova, Zachary Tatlock

Software Verification with ITPs Should Use Binary Code Extraction to Reduce the TCB
 ITP 2018: International Conference on Interactive Theorem Proving
 Ramana Kumar, Eric Mullen, Zachary Tatlock, Magnus O. Myreen

Verifying that Web Pages have Accessible Layout
 PLDI 2018: Conference on Programming Language Design and Implementation
 Pavel Panchekha, Adam Geller, Michael D. Ernst, Zachary Tatlock, Shoaib Kamil

Finding Root Causes of Floating Point Error
 PLDI 2018: Conference on Programming Language Design and Implementation
 Alex Sanchez-Stern, Pavel Panchekha, Sorin Lerner, Zachary Tatlock

Programming and Proving with Distributed Protocols

POPL 2018: Symposium on the Principles of Programming Languages
Ilya Sergey, James R. Wilcox, Zachary Tatlock

Æuf: Minimizing the Coq Extraction TCB
CPP 2018: Certified Programs and Proofs
Eric Mullen, Stuart Pernsteiner, James R. Wilcox, Zachary Tatlock, Dan Grossman

9. Professional development

None

Taylor, Michael Bedford

1. Education
 - Ph.D., Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2007
 - S.M., Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1999
 - A.B., Computer Science, Dartmouth College, 1996
2. Academic experience
 - Associate Professor w / Tenure Joint Appointment, School of CSE and Department of Electrical Engineering, University of Washington, 2017-
 - Associate Professor w / Tenure Director, UCSD Center For Dark Silicene, 2012-2017
 - Assistant Professor, University of California, San Diego, 2006-
 - Research assistant, MIT, 1997-2006
3. Non-academic experience
 - Visiting Research Scientist, Google, 2017
 - Senior Engineer and Consultant, ChipWrights, Inc, 2000-2001
 - Software Engineer, Connectix Corporation, 1996
 - Software Engineer / Intern, Apple Computer, 1995-1996
4. Current membership in professional organizations
 - Institute of Electrical and Electronics Engineers, IEEE
 - Association for Computing Machinery, ACM
5. Honors and awards
 - Selected for IEEE Micro Special Issue on Hotchips, 2018 for *Celerity: An Open Source RISC-V Tiered Accelerator Fabric.*
 - Selected for IEEE Micro Top Picks 2017 for *ASIC Clouds: Specializing the Datacenter.*
 - National Science Foundation CAREER Award, 2009.
 - Selected for IEEE Micro Special Issue on Hotchips, 2011 for *GreenDroid: A Mobile Application Processor for Future of Dark Silicon.*
 - Intel Foundation PhD Fellowship, 2003.
 - Selected for IEEE Micro Special Issue on Hotchips, 2002 for *The Raw Processor: A Scalable 32-bit Fabric for Embedded and General Purpose Computing.*
6. Service activities (within and outside of the institution)
 - *Program Chair, HOTCHIPS 2017, General Chair, HPCA 2015, Theme Co-Leader, Platforms, CFAR 2015, Guest Editor, IEEE Micro 2013, Organizing Committee Member, DASI 2013, WDDD 2012, External Review Committee, PPOPP 2012, Proposal Reviewer, SNSF 2010, Panelist for Proposal Review, NSF MANY, Local Arrangements Chair, ISCA 2007, Reviewer, MANY, Panel Chair, CAW 2002, Program Committee Member, ISCA 2018, TopPicks 2018, MICRO 2017, ISCA 2017, ASPLOS 2016, HOTCHIPS 2016, ISCA 2016, TopPicks 2016, CGO 2015, HOTCHIPS 2014, HPCA 2014, DASI 2013, WDDD 2013, CASES 2013, TopPicks 2013, WDDD 2012, DASI 2012, PACT 2012, CASES 2012, IGCC 2012, FPT 2012, HOTI 2011, CASES 2011, NOCS 2011, MICRO 2010, IPDPS 2010, NOCS 2010, NOCS 2009, ISCA 2007, UCSD Teaching Faculty Recruiting Committee, 2016, UCSD Faculty Recruiting Committee, 2014-2015, UCSD MS Committee, 2013-2014, UCSD ACM Programming Team Coach, 2006-2007, 2009-2016, UCSD Fall Programming Contest World Finals, Faculty Organizer, 2006-2007, 2009-2015, UCSD Spring Programming Contest, Faculty Organizer, 2006, 2010, UCSD Computing Committee, 2008, UCSD Computer Engineering Committee, 2007, UCSD CSE Architecture Competency Evaluator, 2007-2012, UCSD GradComm Graduate Student Advisor, 2006.*

7. Most important publications and presentations from the past five years
- Davidson, S., Xie, S., Torng, C., Al-Hawai, K., Rovinski, A., Ajayi, T., Vega, L., Zhao, C., Zhao, R., Dai, S., Amarnath, A., Veluri, B., Gao, P., Rao, A., Liu, G., Gupta, R., Zhang, Z., Dreslinski, R., Batten, C. and Taylor, M. (2018). The Celerity Open-Source 511-Core RISC-V Tiered Accelerator Fabric: Fast Architectures and Design Methodologies for Fast Chips. *IEEE Micro*, 38(2), pp.30-41.
 - Taylor, M. (2017). The Evolution of Bitcoin Hardware. / *IEEE Computer Society*, pp.58-65.
 - Khazraee, M., Gutierrez, L., Magaki, I. and Taylor, M. (2017). Specializing a Planet's Computation: ASIC Clouds. *IEEE Micro*, 37(3), pp.62-69.
 - Khazraee, M., Zhang, L., Vega, L. and Taylor, M. (2017). Moonwalk. *ACM SIGPLAN Notices*, 52(4), pp.511-526.
 - Khazraee, M., Magaki, I., Vega Gutierrez, L. and Taylor, M. (2017). ASIC Clouds: Specializing the Datacenter. *IEEE Micro*, pp.1-1.

Tompa, Martin

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.
2009-13, 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
5. Current membership in professional organizations
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. Important publications and presentations from the past five years
9. Briefly list the most recent professional development activities

Wang, Xi

1. Education – degree, discipline, institution, year

Ph.D., Computer Science, Massachusetts Institute of Technology, 2014
M.Eng., Computer Science, Tsinghua University, 2008
B.Eng., Computer Science, Tsinghua University, 2005

2. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

University of Washington, Assistant Professor, 2014–2019

3. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

4. Certifications or professional registrations

5. Current membership in professional organizations

ACM member

6. Honors and awards

NSF CAREER Award
Best Paper Award, OSDI 2016
Best Paper Award, SOSP 2013

7. Service activities (within and outside of the institution)

Program Committee: HotOS 2019, OSDI 2018, SYSTOR 2018, SOSP Poster 2017, SOSP Student Research Competition 2017, APSys 2017, USENIX Security 2017, WWW 2017, PLAS 2016, APSys 2016, SYSTOR 2016, EuroSys Doctoral Workshop 2016, USENIX ATC 2015, PLOS 2015

Undergraduate Admissions Committee, 2018, 2019

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

Nickel: A Framework for Design and Verification of Information Flow Control Systems. Helgi Sigurbjarnarson, Luke Nelson, Bruno Castro-Karney, James Bornholt, Emina Torlak, and Xi Wang. In Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Carlsbad, CA, October 2018.

MultiNyx: A Multi-Level Abstraction Framework for Systematic Analysis of Hypervisors. Pedro Fonseca, Xi Wang, and Arvind Krishnamurthy. In Proceedings of the 13th ACM EuroSys Conference, Porto, Portugal, April 2018.

Hyperkernel: Push-Button Verification of an OS Kernel. Luke Nelson, Helgi Sigurbjarnarson, Kaiyuan Zhang, Dylan Johnson, James Bornholt, Emina Torlak, and Xi Wang. In Proceedings of the 26th ACM Symposium on Operating Systems Principles (SOSP), Shanghai, China, October 2017.

An Empirical Study on the Correctness of Formally Verified Distributed Systems. Pedro Fonseca, Kaiyuan Zhang, Xi Wang, and Arvind Krishnamurthy. In Proceedings of the 12th ACM EuroSys Conference, Belgrade, Serbia, April 2017.

Push-Button Verification of File Systems via Crash Refinement. Helgi Sigurbjarnarson, James Bornholt, Emina Torlak, and Xi Wang. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, GA, November 2016.

Investigating Safety of a Radiotherapy Machine Using System Models with Pluggable Checkers. Stuart Pernsteiner, Calvin Loncaric, Emina Torlak, Zachary Tatlock, Xi Wang, Michael D. Ernst, and Jonathan Jacky. In Proceedings of the 28th International Conference on Computer Aided Verification (CAV), Toronto, Canada, July 2016.

Specifying and Checking File System Crash-Consistency Models. James Bornholt, Antoine Kaufmann, Jialin Li, Arvind Krishnamurthy, Emina Torlak, and Xi Wang. In Proceedings of the 21st International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Atlanta, GA, April 2016.

A Differential Approach to Undefined Behavior Detection. Xi Wang, Nickolai Zeldovich, M. Frans Kaashoek, and Armando Solar-Lezama. Communications of the ACM, 59(3):99–106, March 2016.

9. Briefly list the most recent professional development activities

Weld, Daniel S.

Education

Ph.D. Artificial Intelligence	1988
Massachusetts Institute of Technology	
M.S. Computer Science	1984
Massachusetts Institute of Technology	
B.S. Computer Science (Cum Laude & Highest Honors)	1982
B.A. Molecular Biophysics and Biochemistry	
Yale University	

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

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.

Certifications

Membership in Professional Organizations

Association for the Advancement of Artificial Intelligence (AAAI), Fellow
Association of Computing Machinery (ACM), Fellow

Selected Honors

Entrepreneurial Faculty Fellow	2012
University of Washington	
Honorable Mention – Best Student Paper	2008
<i>17th International World Wide Web Conference (WWW-08)</i>	
Best Paper	2008
<i>ACM Conference on Human Factors in Computing Systems (CHI 2008)</i>	
Best Paper	2007
<i>16th ACM Conference on Information and Knowledge Management (CIKM 2007)</i>	
Fellow 2006–	
Association of Computing Machinery	
Fellow	
1999–	
American Association of Artificial Intelligence	
WRF / T. J. Cable Endowed Professorship	1999–
University of Washington	
Young Investigator Award	1990
Office of Naval Research	

Presidential Young Investigator Award National Science Foundation	1989
John E. Bierwirth Scholarship Yale University	1979-82
Wilfred Freeman Fellowship Phillips Academy	1978

Professional Service

Advisory & Review Boards

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)
 Councillor, American Association of Artificial Intelligence (AAAI) (1994-1997)
 Member of Advisory Board, *Journal of Artificial Intelligence Research* (1992-)
 Member and Editor, AAAI/NSF Committee on Intelligence in the NII (1994)
 Member, AAAI/ARPA Committee on Twenty-First Century Intelligent Systems (1994)

Editorial Activities

Area Editor, *Journal of the ACM* (2007-2012)
 Editorial Board Member, *Artificial Intelligence* (1999-2008)
 Guest Editor, *Artificial Intelligence* special issue on Intelligent Internet Systems (1998)
 Associate Editor, *Journal of Artificial Intelligence Research* (1993-1996)
 Guest Editor, *Computational Intelligence* special issue on qualitative reasoning (May 1992)

Program Committee Chair

IUI Program Chair (2009)
 IUI Associate Chair (2007)
 AAAI (1996), Senior-Member Paper Track (2006), Challenge Track (2010)
 AGENTS (1998, Area Chair for Software Agents)
 International Workshop on Qualitative Reasoning (1993)

Program Committee Member

AAAI (1990, 1991, 1992, 1993, 1996 (chair), 1999 (senior), 2000 (senior), 2005 (senior), 2006 (track co-chair), 2008 (senior), 2010 (track co-chair)
 HCOMP (2013)
 ICAPS/AIPS/ECP (1995, 1997, [1998, 200, 2002 (Planning Comp Rules Committee)], 2003, 2004, 2005, 2006, 2007 Workshop Chair, 2008)
 IJCAI (Area Chair, 2009)
 UIST (2009)
 WWW (2001, 2012)

Reviewer

Artificial Intelligence
Journal of Artificial Intelligence Research
Cognitive Science
Machine Learning
Journal of the ACM
International Journal of Intelligent Systems
International Journal for Artificial Intelligence in Engineering
IEEE Transactions on Systems, Man and Cybernetics
IEEE Pattern Analysis and Machine Intelligence

ACM Transactions on Software Engineering and Methodology
National Science Foundation

Academic Service, Departmental

Curriculum Committee, 1993-1995, 2001-2002
Executive Committee, 1993-1995, 2003-2004, 2005-2006, 2008-2009, 2011-12
Faculty Recruiting Committee, 1990-1991, 1992-1993, 1996-1998 (chair), 2002-2004 (chair)
Graduate Admissions Committee, 2002, 2006, 2011 (chair), 2012
Graduate Program Coordinator, 1998-1999
Graduate Review of Progress, 2005-2007
Lab Policy Committee, 1989-1994.
Undergraduate Admissions Committee, 1992

Academic Service, University wide

Electrical Engineering 10-Year Review Committee, 2011-12
College of Engineering Promotion & Tenure, 2007-2010 (chair 2009-10)
Computer Science and Engineering Chair Search Committee, 1992-1993

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.

Recent Professional Development

Curriculum development for CSE 446 *Machine Learning*, CSE 454 *Advanced Internet and Web Services* & CSE 573 *Artificial Intelligence I*

Zahorjan, John

1. Education

PhD Computer Science, University of Toronto, 1980
Computer Science, University of Toronto, 1976
Mathematics, Brown University, 1975

M.Sc.
B.Sc. Applied

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:

College of Engineering Promotion and Tenure Committee, 2015-present
CSE 100-Level Course Review Committee, 2018-19

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.

Appendix C – Equipment

Student Computing Resources (physically accessible)

General Purpose Instructional Laboratories

- 140 Dell or Lenovo AIO Workstations
- 10 Dell Precision Workstations with GPU cards
- 240 LCD displays (24", 27", 40")

Animation Instructional Laboratory

- 29 Dell or HP or Custom-built Workstations
- 56 LCD displays (20" or 24")

Hardware Capstone Laboratory

- 18 Dell Precision Workstations
- 30 LCD Displays (20" and 24")
- 12 Power Supplies (Tektronix)
- 7 Multimeters (BK Precision)
- 14 Function Generators (Tektronix)
- 11 Oscilloscopes (Keysight/Rigol/Agilent)

Other Capstone Laboratories

- 32 Dell AIO Workstations
- 10 GPU enabled, custom built workstations
- 40 LCD displays (24" and 27")

Printers

- 5 Ricoh Multi-function Mono Printers

Instructional Staff Computing

- 70 Faculty Desktops
- 250 Graduate Student Desktops/Laptops (acting in a TA role)

Instructional Shared Facilities Hosted in Datacenter

Instructional Cycle Cluster (8 servers; provides remote access to course software)

- 4 Dell PowerEdge R430s (2x 12C Intel Xeon E5-2670v3 CPUs 2.3GHz, Total 48 cores and 128GB RAM per node)
- 4 Supermicro servers (2x14C Intel Xeon Gold 6132 CPUs 2.6GHz, Total 56 cores and 192GB RAM per node)

Instructional Code Repository/Runner Services, Gitlab (2 servers)

- 1 Dell PowerEdge R630 32GB RAM, 2.80 GHz, 5TB Storage Capacity
- 1 Dell PowerEdge R710 24GB RAM, 2.66 GHz

Instructional File Server (1 server; student home directories):

- Dell PowerEdge R510, 32GB RAM, 18TB Storage Capacity

Instructional Virtualization Infrastructure (8 servers, course project servers, VDI)

6 Dell PowerEdge R710s, dual 6-core 2.4Ghz processors, 24GB RAM

1 Dell PowerEdge R430 (2x 12C Intel Xeon E5-2670v3 CPUs 2.3GHz, Total 48 cores and 192GB RAM per node)

1 Dell PowerEdge R430 (2x 12C Intel Xeon E5-2670v3 CPUs 2.3GHz, Total 48 cores and 128GB RAM per node)

Instructional Web Server (1 server; course webs, course projects, course archives)

Dell PowerEdge R520, 32GB RAM, 18TB Storage Capacity

Animation Renderfarm (6 servers, support back end animation rendering workloads)

5 Dell PowerEdge R420, 192GB RAM, dual 2.3GHz processors

1 Dell PowerEdge R430, 128GB RAM, dual 2.3GHz processors

Appendix D – Institutional Summary

1. The Institution

- a. Name and address of the institution
University of Washington, Seattle, WA 98195
- b. Name and title of the chief executive officer of the institution
Ana Mari Cauce, President
- c. Name and title of the person submitting the Self-Study Report.
Professor Arvind Krishnamurthy
- d. Name the organizations by which the institution is now accredited, and the dates of the initial and most recent accreditation evaluations.
 - Northwest Commission on Colleges and Universities (NWCCU)
 - Initial accreditation – 1918
 - Most recent accreditation evaluation – Accreditation was reaffirmed in January 2014, after a comprehensive self-evaluation report and site visit in autumn 2013.

2. Type of Control

State-assisted Public Research University

3. Educational Unit

The Computer Engineering Program is administered by the Paul G. Allen School of Computer Science & Engineering which is an academic unit within the College of Engineering. The College of Engineering is a separately organized unit with its own budgetary and program control within the University of Washington. The Director of the Paul G. Allen School of Computer Science & Engineering is Hank Levy. His term is up at the end of June and a new Director will be placed in the role. Hank Levy reports to the Dean of the College of Engineering, Mike Bragg (Note: Dean Bragg will be stepping down from the Dean position effective June 30, 2019). Dean Bragg reports to the Provost and Executive Vice President, Mark Richards. University of Washington and College of Engineering organizational charts are provided in Figures D-1 and D-2.

Figure D-1 University of Washington Organization Chart

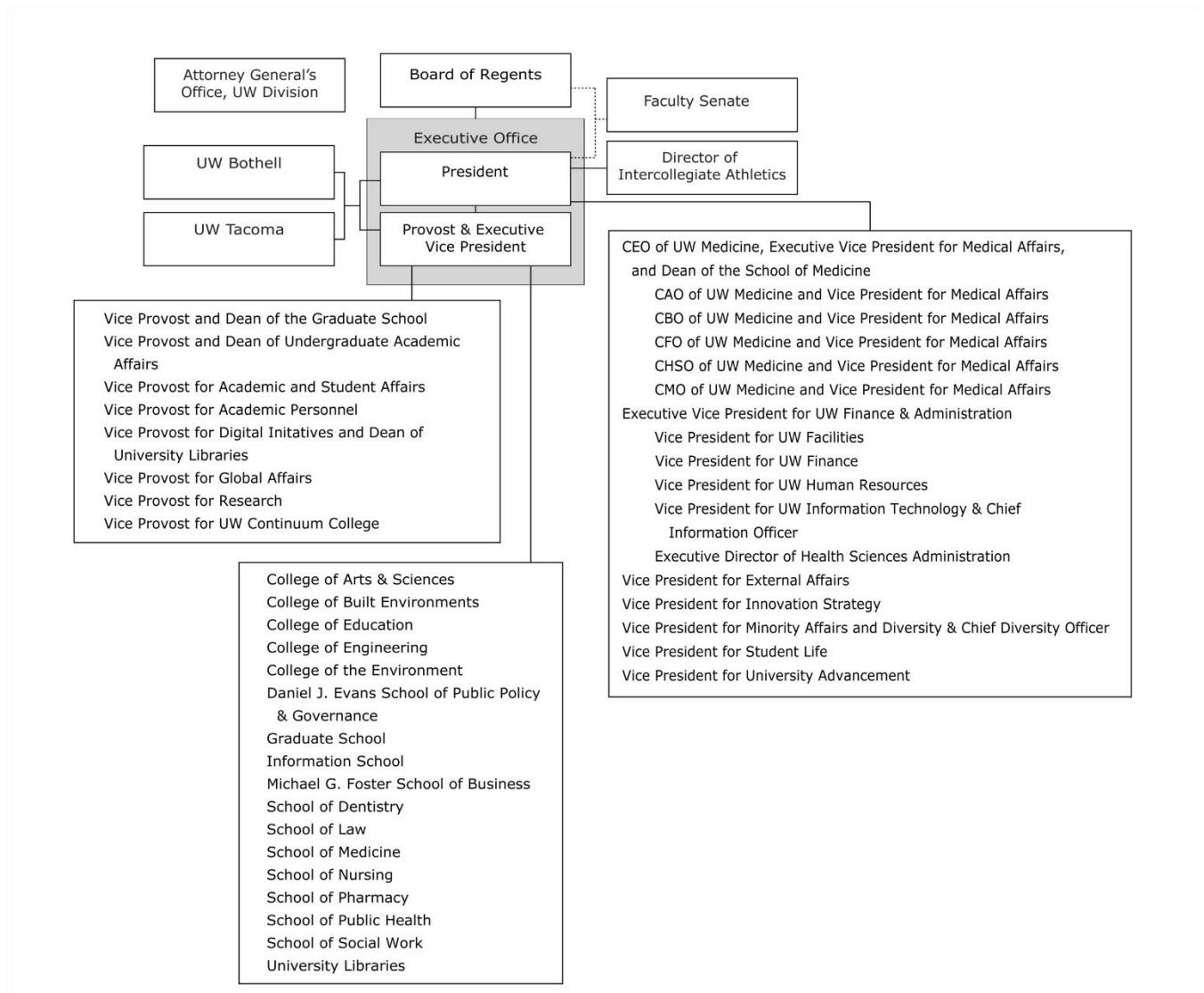
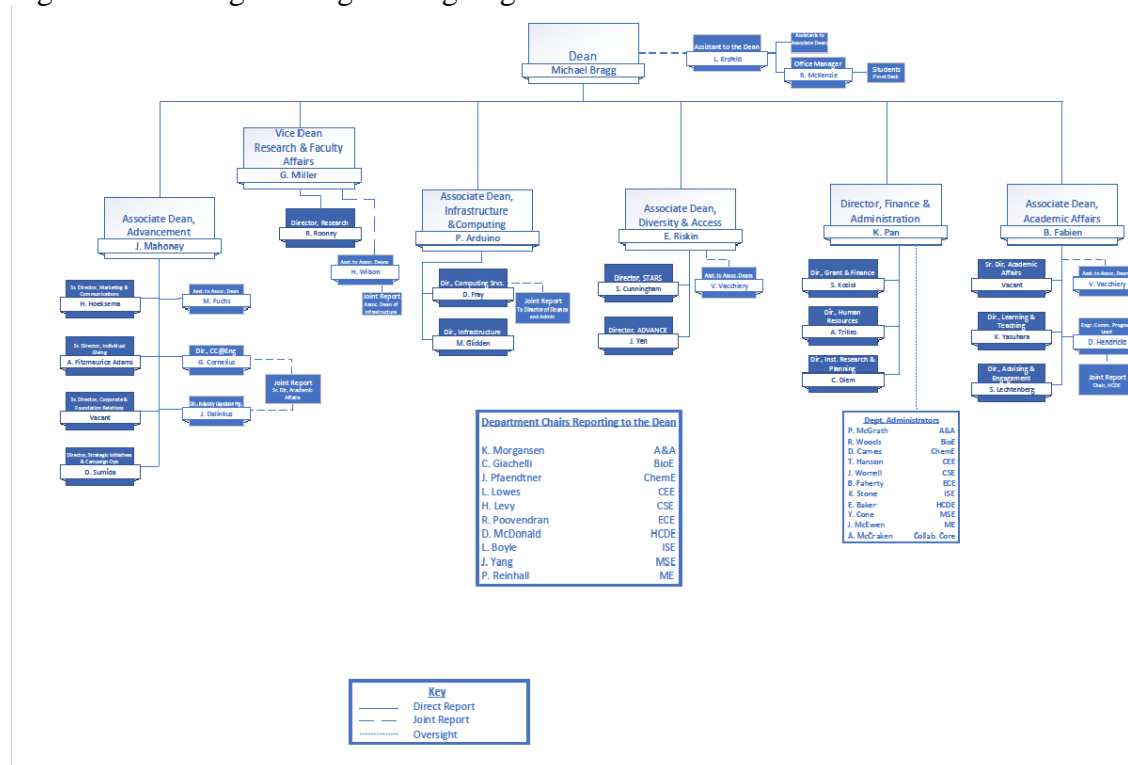


Figure D-2 College of Engineering Organization Chart



4. Academic Support Units

The names and titles of the individuals responsible for the units that teach courses required by the program are provided below.

Unit	Head
Applied Mathematics	Bernard Deconinck, Professor and Chair
Biology	H.D. 'Toby' Bradshaw, Professor and Chair
Chemistry	Michael Heinekey, Professor and Chair
English	Anis Bawarshi, Professor and Chair
Mathematics	John Palmieri, Professor and Chair
Physics	Blayne Heckel, Professor and Chair
Statistics	Thomas Richardson, Professor and Chair

5. Non-academic Support Units

The names and titles of the individuals responsible for each of the units that provide non-academic support to the program being evaluated, e.g., library, computing facilities, placement, tutoring, etc. are provided below.

Unit	Head
------	------

UW Enrollment Management	Philip Ballinger, Associate Vice Provost
UW Registrar	Helen Garrett, University Registrar and Chief Officer for Enrollment Information Services
UW Information Technology	Aaron Powell, Vice President for UW Information Technology and Chief Information Officer
UW Libraries	Lizabeth (Betsy) Wilson, Vice Provost for Digital Initiatives and Dean of University Libraries
UW Office of Minority Affairs & Diversity	Rickey Hall, Vice President for Minority Affairs & Diversity
College of Engineering Academic Affairs	Brian Fabien, Professor and Associate Dean
Engineering, Learning, & Teaching	Jim Borgford-Parnell, Director
College of Engineering Office of Diversity & Access	Eve Riskin, Professor and Associate Dean
College of Engineering Computing Services	David Fray, Director
Career Center @ Engineering	Gail Cornelius, Director
Engineering Library	Mel DeSart, Head, Engineering Library
UW Office of First Year Programs	LeAnne Jones Wiles, Director
UW ADVANCE Center for Institutional Change	Eve Riskin, Faculty Director

6. Credit Unit

The University of Washington operates on a quarter system. The standard academic year consists of three 10-week quarter terms. Each quarter also includes a final exam week. For a traditional lecture-based class, one quarter credit normally represents one class hour and two hours of work outside of class per week. Contact hours include different formats including laboratories, quiz session, and tutorial sessions. Depending on the expectation for work outside of class, one quarter credit normally represents two to three hours of lab, quiz, or tutorial per week. Considering the diversity of teaching styles and learning environments, the University and the College of Engineering curriculum committees will consider approving courses that vary from the standard guidelines. For such courses, faculty must present written justification for the departure from the standard course credit/contact hour ratios.

7. Tables

Complete the following tables for the program undergoing evaluation

Table D-1. Program Enrollment and Degree Data

Computer Engineering

	Academic Year	Enrollment Year					Total Undergrad	Total Grad	Degrees Awarded			
		1st	2nd	3rd	4th	5th			Associates	Bachelors	Masters	Doctorates
Current Year	FT	1	10	53	67		131	324				
	PT	0	0	3	10		13	183				
1 year prior to current year	FT	15	16	55	56		142	260		36	100	20
	PT			1	12		13	171				
2 years prior to current year	FT	20	19	43	58		140	241		33	107	18
	PT				11		11	145				
3 years prior to current year	FT	6	15	37	72		130	241		50	103	16
	PT			3	9		12	148				
4 years prior to current year	FT	3	14	45	83		145	226		61	104	25
	PT			1	19		20	149				

Give official fall term enrollment figures (head count) for the current and preceding four academic years and undergraduate and graduate degrees conferred during each of those years. The "current" year means the academic year preceding the on-site visit.

FT—full-time
PT—part-time

Table D-2. Personnel

Computer engineering

Year: Fall 2018

	HEAD COUNT		FTE ¹
	FT	PT	
Administrative ²	42	6	45.7
Faculty (tenure-track) ³	69	1	66
Other Faculty (excluding student Assistants)	10	0.5	10.5
Student Teaching Assistants ⁴	154	132	286
Technicians/Specialists	47	4	49.1
Office/Clerical Employees	10	0	10
Others ⁵	0	0	0

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).
5. Specify any other category considered appropriate, or leave blank.

Submission Attesting to Compliance

Only the Dean or Dean's Delegate can electronically submit the Self-study Report.

ABET considers the on-line submission as equivalent to that of an electronic signature of compliance attesting to the fact that the program 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*.