

Computer Science & Engineering Handbook

Department of Computer Science & Engineering

FOR UNDERGRADUATES

2000-02

Contents

3 Letter from the Chair

Information for Prospective Students

5 Computer Science and Computer Engineering Degree Information

6 Frequently Asked Questions

Admissions

9 Prerequisites for Admission

10 Application Instructions

11 Selection Process

Information for Majors

CURRICULUM

13 Degree Requirements

14 CSE Senior Electives

15 Academic Advising for CSE Majors

ADDITIONAL OPPORTUNITIES

16 Co-op Program

16 Center for Career Services

16 Research Opportunities

16 Scholarships

16 Teaching Assistantships

17 Honors Programs

POLICIES AND PROCEDURES

18 Satisfactory Progress

18 Review of Progress

19 Academic Misconduct

19 Graduation

19 Applicability

Course Listings

21 Introductory Courses

22 Courses for Majors

25 Courses for Non-Majors

27 Computer Science and Engineering Faculty

Letter from the Chair

Thank you for your interest in the University of Washington's Department of Computer Science & Engineering!

Of course, it's great to major in a field that's burgeoning, a field in which the world's most vibrant industry is right here in the Puget Sound region (where we boast more than 2,000 software and digital media firms). But the right reason to choose Computer Science or Computer Engineering as your major is that these fields are where the intellectual excitement is, where the breakthroughs are occurring, where the smartest young men and women are choosing to work. Computer Science and Computer Engineering are where the action is - from robots to personal communication systems to intelligent agents to computer animation to computational biology.

UW's Department of Computer Science & Engineering is ranked among the top ten in the nation, along with Stanford, MIT, Berkeley, Carnegie Mellon, Cornell, Princeton, Texas, Illinois, and Wisconsin. As an undergraduate in our department you'll have very special opportunities - the chance to reach, and even to help define, the forefront of the field. You'll find faculty, staff, and students who are among the finest at the University of Washington. You'll find state-of-the-art labs and curriculum. You'll find junior and senior classes that typically have less than 50 students, affording close faculty contact. You'll find co-ops and internships that complement your education; laboratory assistantships and teaching assistantships; "capstone" design courses where you'll work in a team to design and implement significant systems; and the chance to participate actively in research alongside graduate students and faculty.

We're committed to helping you get the great education that you deserve. We hope you choose to join us.

Sincerely,

David Notkin
Professor and Chair



The background of the slide is a dark red color. It features a pattern of white binary code (0s and 1s) arranged in diagonal lines across the top half. The bottom half of the slide has a dark red background with a subtle, lighter red grid pattern. A dark blue horizontal bar is positioned behind the title text.

Information for Prospective Students

There is a wide variety of career opportunities available in computing. Computer science and computer engineering are just two of these. You can find information about job fields in the more general Computing and Information Technology fields and ways to prepare for them at

<http://www.cs.washington.edu/info/prospective/ITCareers.html>.

Computer Science and Computer Engineering Degree Information

The department offers two undergraduate degrees: a B.S. in Computer Engineering (CompE) and a B.S. in Computer Science (CS). These degree programs have much in common in that they share a common core of fundamental courses both in the prerequisites for admission as well as the 300-level courses in the Department. They both require a total of 180 credits. The differences lie in the requirements beyond the common core and in courses outside the department.

The computer engineering curriculum includes a background in engineering principles, engineering design, mathematics, and electrical engineering which provide a basis for understanding the underlying technologies of modern computing systems and the tradeoffs faced by designers. In addition, the program emphasizes written and oral communication skills.

The computer science curriculum includes a wider range of social science/humanities requirements in line with other programs in the College of Arts and Sciences. Students have a wider choice in shaping their program and can concentrate in topics such as applied mathematics and statistics, biology and medicine, ethics and law, business and economics, and many others.

Within the department, the two degree programs also share many upper-level courses. Given a common core, a computer science student can shape their program to include many of the courses required for the computer engineering degree. The same is true for a computer engineering student. Furthermore, the computer engineering program has two options: one focusing on hardware design while the other lies closer to computer science and concentrates on embedded software.

Computer scientists focus on software systems and understanding the limits of computation within the context of real and abstract computing devices. Computer engineers tend to focus more on the design of embedded systems (systems that involve both hardware and software components). As these systems often involve constraints on cost, size, and/or power consumption, designers are often faced with tradeoffs to make that will satisfy competing requirements.

Computer engineers design and implement computer hardware and software systems to solve a variety of problems in diverse areas such as communications, manufacturing, robotics, computer graphics, databases, and many others. A computer engineer might be involved in fundamental research, hardware design and fabrication, software creation, or systems integration.

Computer scientists are generally involved in applications that require the design of the representation of information; methods for its storage and retrieval; algorithms to display, edit, and transform the information; programming and mathematical languages to express the algorithms that operate on the information; and the hardware and software to execute these algorithms.

Both degree programs prepare students to be highly successful in their technical careers whether they go on to positions in industry or continue immediately on to graduate school.

Frequently Asked Questions

What is the Difference between Computer Science and Computer Engineering?

Computer Engineering and Computer Science are closely related. The department's junior-level courses are required of both CompE and CS majors, and many of the senior courses are taken by students in both disciplines. However, there are differences between the two degree programs. The key differences are as follows:

CompE students must satisfy the College of Engineering distribution requirements; CS students must satisfy the College of Arts and Sciences distribution requirements. The CompE major has more required courses and fewer electives. CompE students take more courses in computer hardware and electrical engineering than CS. The CompE major is appropriate for students interested in building systems that include both custom hardware and software components, who have interests in EE, or who specifically want an Engineering degree.

The flexibility of the CS major allows students to double major with another Arts and Sciences program (for example, Mathematics or Economics). Please see the previous section (Degree Information) for additional details.

What can I do with a degree in Computer Science & Engineering?

Both CS and CompE graduates have had excellent success in finding positions after graduation. Many positions are equally suitable for graduates with either degree.

We train our students for two career paths—graduate work in CS&E and entry into industry. Both programs are academically rigorous, and our graduates are highly competitive at the top graduate schools in the country. They are also at the forefront of the practical developments in computer technology and applications, as well as of theoretical investigations of computers, algorithms, and data. These qualities make our students highly valuable to local and national industry.

Can I major in both degree programs?

Due to the large overlap, a double major of Computer Engineering with Computer Science is not practical. Both degree programs are competitive and have more qualified applicants than spaces available.

Can a student attend part-time?

We offer a full-time, day program only. Students admitted to the program must pursue the degree on a full-time basis.

Can I take some of the prerequisite courses at a community college?

Many students who apply to the department are transfer students or have taken some courses at a community college. For information about UW Course Equivalencies at Washington State Community Colleges, please see the following website: http://www.washington.edu/students/uga/transfer/course_equiv.html

Is there a minimum grade point average for an application to be considered?

We do not automatically eliminate any application because of a low grade point average. On the other hand, it is generally the case that the higher one's grades, the better chance one has of being accepted. It is also the case that in a given year a large number of applicants to CSE present very strong academic credentials.

What weights are attached to grades, work experience, essay and so forth?

We don't assign specific points to each of these things. We look at each application as a whole and attempt to assess the overall qualities of each applicant. For instance, the essay gives us some notion of how an applicant writes and thinks, but the questions themselves do not have right or wrong answers. Please see the essay question (page 10) for some ideas about how to tell us about yourself. Each application receives multiple readings. We ask ourselves how well a student has used the resources available to him or her, realizing that some students will have had more or fewer opportunities available to them than others. If we had to state one factor as being more important than others, it would be the applicant's performance in the prerequisite courses.

What is the Early Decision program?

We enroll up to 15% of our incoming class directly out of high school. Freshmen applicants to the University of Washington listing Computer Science or Computer Engineering as their intended major, and who are Washington State residents, will be automatically considered. Competitive applicants will have taken or be taking calculus. Physics is the preferred high school laboratory science course.

What are the Deadlines for Applying?

We admit students in autumn and spring quarters. The deadline for autumn quarter is JULY 1. The deadline for spring quarter is FEBRUARY 1. Please see the section on Admissions for additional information about applying.

How can I learn more?

We recommend you attend an information session about the programs. These sessions are held regularly, and the schedule is available at our website (<http://www.cs.washington.edu/education/info.sessions.html>) or by calling 206-543-1695.

Admissions



Admission Prerequisites

Students applying to both the CS and CompE Programs must complete the following requirements prior to the deadline date:

- Be enrolled as a student at the UW or apply to enroll through the Office of Admissions, 320 Schmitz Hall (206-543-9686; askuwadm@u.washington.edu)
- Have a cumulative GPA (including all transfer units) of at least 3.0 (2.5 for CompE)
- Complete the following courses:
Math 124, 125, 126 or Math 127, 128, 129 (Calculus)
Physics 121 (Mechanics)
CSE 142 (Computer Programming I)

Additional Computer Science Prerequisites:

CSE 143 (Computer Programming II)

Earn a minimum of 45 quarter credits applicable toward a bachelor's degree

Additional Computer Engineering Prerequisites:

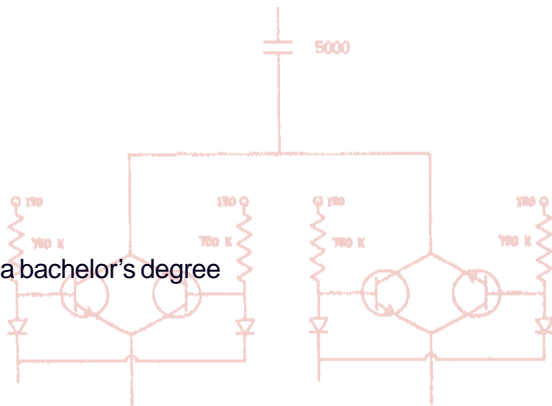
Math 307 (Differential Equations)

Physics 122;123 (Electromagnetism & Oscillatory Motion; Waves)

Chemistry 142 (General Chemistry)

Five credits of English composition (selected from English 111, 121, 131, 197, 198, 199, or Comparative Literature 240)

Earn a minimum of 64 quarter credits applicable toward a bachelor's degree



Early Admission (Computer Engineering Only)

Students eligible for early admission must be currently enrolled at the UW, have completed 15 credits of Math 124, 125, 126 or higher, 10 credits of Physics 121, 122, 123 or Chem 142, CSE 142, 5 credits of English composition, and have completed at least 15 credits at the UW.

This option is available for autumn quarter admission only.

Early Decision

The UW Department of Computer Science & Engineering enrolls up to 15% of its incoming class directly out of high school. Freshman applicants to the University of Washington listing Computer Science or Computer Engineering as their intended major, and who are Washington State residents, will be automatically considered. Competitive applicants will have taken or be taking calculus. Physics is the preferred high school laboratory course.

Application Instructions

We welcome your application to the CSE department. Admission is based upon, but not limited to, evaluation of the following factors: (1) overall scholastic record; (2) grades in CSE prerequisite courses; (3) written communication skills; (4) evidence of computing ability and promise of achievement in the information technology field.

Application deadlines:

July 1 - autumn quarter

February 1 - spring quarter

1 COMPLETED APPLICATION

The Online Application is available at <http://www.cs.washington.edu/admissions>

Transfer students must also be admitted to the University of Washington through the Office of Undergraduate Admissions, 320 Schmitz Hall. They should indicate their interest in Computer Engineering and/or Computer Science on their UW application and submit the application to the University Admissions Office early enough to meet both University and Department deadlines.

2 STATEMENT OF PURPOSE

The personal statement is an important part of your application for admission. Your official transcript provides one, somewhat narrow view of you, but it cannot show many other qualities of character and experience that we consider equally important. The personal statement is your best way to show us “the whole person. The following are examples of some topics that you may wish to discuss in your statement. These are only examples, and not all examples will apply to all applicants. You should discuss whatever topics are most appropriate for conveying to us the richness of who you are. It is helpful to be as specific and concrete as you can about your activities, your background and experiences, and any challenges you faced.

- While good grades are important, equally important to us is that we have a rich, diverse student population. If you have had previous experiences, or have an unusual or interesting background or culture, or have current activities or interests that you think would enrich our community or make it more diverse, then please discuss them in your statement.
- If you have had to overcome any significant obstacles, difficulties, or challenges to reach where you are now, including medical, financial, cultural, and personal challenges, please discuss them and the ways in which they affected you.
- We are very interested to hear about your future professional plans, goals, and aspirations, and in how getting a CS or CE degree fits into your plans.
- If you wish to transfer from another four-year institution before graduating, then please let us know why you want to transfer to our department.
- If you have applied previously to our department, then please point out what has improved in your application since the last time you applied.

A typical statement of purpose is one page in length.

3 COPIES OF ALL UW AND NON-UW TRANSCRIPTS (Unofficials from Gateway Center Advising are acceptable, but NOT those from MyUW)

These transcripts are in addition to those submitted to the University of Washington's Admissions Office. Transcripts for the most recent quarter (spring for autumn admissions and autumn for spring admissions) must be included. We will accept unofficial UW transcripts available from the Gateway Center in Mary Gates Hall.

Students formerly majoring in Computer Engineering or Computer Science at the University of Washington and wishing to re-enroll should consult with an advisor.

Selection Process

Computer Science: Computer Science accepts approximately 80 students per year and is offered through the College of Arts and Sciences.

Computer Engineering: Computer Engineering accepts approximately 80 students per year and is an ABET-accredited program offered through the College of Engineering.

The Admissions Committee evaluates the qualifications of applicants exclusively on the basis of written material submitted by each applicant. Due to the large number of applications, it is not possible to conduct personal interviews in the selection process. Students are encouraged to attend an information session prior to submitting their application.

Strong written and oral communication skills are essential to success in CompE and CS, both in the undergraduate programs and later in the workplace or graduate school. For this reason, the statement of purpose and evidence of written and oral communication skills will be evaluated by the Admissions Committee both on the basis of content and as a sample of the student's writing ability.

Enrollment is limited due to space, faculty, and budgetary limitations. Applicants who, in the judgment of the Admissions Committee, are most likely to succeed in the program and contribute to the field will be admitted.

Applicants will be notified of admission status by letter 4 to 6 weeks after departmental application deadlines.

Students may not defer admission. Students not entering in the quarter for which they applied, must reapply for the quarter they wish to enroll.

In conformity with a standing policy of the University, the Department provides equal opportunity in education without regard to race, color, national origin, sex, age or handicap in accordance with Title VI of the Civil Rights Act of 1964, Title IX of the Education Amendments, Section 504 of the Rehabilitation Act of 1973, and the Age Discrimination Act of 1975.

Appeals Procedure

The only legitimate basis for an appeal is that there was an error in the data considered by the Admissions Committee. Incomplete or inaccurate records submitted by the applicant will not be considered as a basis for an appeal. Students denied admission to the program may wish to consult with an undergraduate advisor for additional information and guidance. Students may request a review of the Admissions Committee decision by writing a letter to the Admissions Committee citing the reasons for the appeal. This letter must be submitted within two weeks after the notification of denial is mailed.

Information for Majors

PROGRAM MISSION STATEMENT

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

From our program mission statement, we derive the following four goals for our program:

- **Core knowledge:** To offer a challenging and broad-based education in the fundamental ideas, techniques, and tools underlying the discipline of computer science and engineering.
- **Application:** To impart the ability to apply the core knowledge of the discipline in making the many engineering tradeoffs that are central to solving problems and creating products.
- **Educational renewal:** To prepare our students to continually renew their education in a rapidly developing discipline so that they may realize their full potential throughout their career.
- **Social context:** To foster personal development not only in the social aspects of the discipline itself but also in terms of understanding the impact of our field on society as a whole.

For more information, see <http://www.cs.washington.edu/education/abet>

Computer Engineering and Computer Science Degree Requirements

All Computer Engineering and Computer Science students must complete the following courses
(32 credits): CSE 142, 143, 321, 322, 326, 341, 370, 378

Additional Computer Science requirements:

Must take at least 4 classes from the following (12-16 credits): CSE 401, 403, 421, 431, 444, 451, 455, 457, 461, 471, 473.

CSE Electives (7 credits)

See CSE electives list on page 14 of the handbook

Math (24 credits):

MATH 124, 125, 126, 308, MATH/STAT 390 or 391; any 300-level MATH course (except MATH 300, 354, and 355); or 2 credits of senior electives (STAT/MATH 394 and 395 together may be substituted for the last two requirements).

Science (15 credits):

PHYS 121, 122, 123

General Education (90 credits):

VLPA, I&S, the Natural World (75 credits)

Language Skills (5-20 credits)

Reasoning & Writing (15 credits)

Free Electives (Approximately 30-50 credits)

Additional Computer Engineering requirements:

CSE 451, 461, EE 215 and 233

Hardware Option (18 credits): CSE 467, 471, EE 331; CSE 468 or 477

Software Option (16-17 credits): CSE 403, 466; **one** of CSE 401, 457, or 471; CSE 476 or 481

CSE Electives (13 credits)

See CSE electives list on page 14 of the handbook

Math (25 credits):

MATH 124, 125, 126, 307, 308, MATH/STAT 390

Science (20 credits):

PHYS 121, 122, 123; CHEM 142

General Education Requirements (42 credits)

VLPA and I&S (30 credits):

As prescribed by the College of Engineering

Written and Oral Communication (12 credits):

English Composition, TC 231, 333

Free Electives (13-15 credits; depends on option chosen)

CSE Senior Electives

COMPUTER ENGINEERING & COMPUTER SCIENCE		
CSE		Any graded 400-level majors course (includes CSE 498, not 499)
		Up to 6 credits of CSE 498 (Senior Project) are allowed, except for honors students, who must take 9 credits of CSE 498B.
ENGR/AMath	401,402,403	Methods in Applied Mathematics I,II,III (4,4,4)
Geog	460	GIS Analysis (5)
	463	GIS Workshop (3-5)
	465	Analytical Cartography (5)
IE/TC	455	User Interface Design (3)
Info	311	Organizational, Societal and Global Perspectives on Info Systems (5)
	370	Assessment and Evaluation Techniques (5)
	444	Value-Sensitive Design (5)
	454	Information Policy: Domestic and Global (5)
Ling/Anth	461, 462	Syntax (4,4)
Ling/Phil	479	Formal Semantics and Natural Language (3)
Math	334, 335, 336	Accelerated Advanced Calculus
		402, 403, 404 Introduction to Modern Algebra (3,3,3)
	407	Linear Optimization (3)
	408	Nonlinear Optimization (3)
	409	Discrete Optimization (3)
	414,415	Number Theory (3,3)
	424,425,426	Fundamental Concepts of Analysis (3,3,3)
	435,436	Introduction to Dynamical Systems (3,3)
	438,439	Introduction to Partial Diff Equations (3,3)
	441	Topology (3)
	442	Differential Geometry (3)
	461,462	Combinatorial Theory (3,3)
	464,465, 466	Numerical Analysis I, II, III (4, 4, 4)
Music	401, 402, 403	Computer Music Seminar (3,3,3)
Stat	341,342	Introduction to Prob & Stat Inference I,II (4,4)
	421	Intro to Applied Stat & Experimental Design (4)
Stat/Math	394, 395	Probability I, II (3,3)
	491,492	Introduction to Stochastic Processes (3,3)
TC	401	Style in Scientific and Technical Writing (3)
TC	407	Computer Documentation (3)
COMPUTER ENGINEERING ONLY		
EE	332	Devices and Circuits II (5)
EE	341	Linear Systems Analysis (5)
EE		Any 400 level majors courses except EE 471, EE 472, and EE 478

Courses not on this list can apply toward the senior elective component, subject to approval by the Undergraduate Faculty Advisor.

Academic Advising for CSE Majors

Location: 114 Sieg Hall

Contact: telephone: 206/543-1695

e-mail: ugrad-advisors@cs.washington.edu

Web: <http://www.cs.washington.edu>

$[a^+ \quad ; \quad]$ $[a^+ \quad a^-; \quad]$ $[a^+b^+ \quad b^-a^-; \quad]$ $[a^+b^+c^+c^-b^-a^-; \quad]$ $[a^+b^+c^+c^-b^-a^-; a^-c^+b^+]$

The advising staff is available to help you with many aspects of planning and completing your undergraduate degree program. They can help you:

- plan your degree and quarterly schedules
- assess progress toward your degree
- understand and navigate UW and departmental policies and procedures
- find and apply for scholarships
- resolve problems or refer you to other UW resources that can assist you with academic and personal issues
- participate in UW and departmental programs offering leadership and professional development opportunities such as:
 - co-ops
 - research
 - teaching assistantships
 - community service
 - honors program in CSE
 - study abroad

The advising staff wants to ensure that all CSE majors benefit from the services we offer. We encourage students to contact us when they need assistance and to stay informed about special programs through e-mail and in person. The best ways to do this are to:

- regularly read the uw.cs-ugrads newsgroup
- make regular appointments with an advisor (send email to ugrad-advisor@cs.washington.edu)
- take advantage of drop-in advising hours

Advising for pre-majors:

The first step for pre-majors is to read this CSE Handbook. Then, plan to attend an information session. These sessions are offered on a regular basis during the academic year and the schedule is on our website at www.cs.washington.edu/admissions

Additional Opportunities

Co-op Program

The Co-op Office in 353 Loew Hall coordinates co-op jobs and minority student internships. These programs give students the opportunity to investigate the practice of the profession and define their own career goals by spending one or more quarters in a job in the computer industry while enrolled in the degree program. A co-op job is an opportunity to put academic knowledge to use in a way that helps students see the relationships between various parts of the curriculum and add to that knowledge in important ways. Students gain a wide variety of skills and abilities related to working in a team setting — a setting that is very difficult to emulate in a university. Many Computer Science & Engineering students have found co-ops to be a valuable part of their studies, and majors are strongly encouraged to take advantage of the opportunities that co-ops present for professional and personal development.

Students going on co-op should notify an undergraduate advisor and plan a tentative schedule for their returning quarter. The quarters they spend off campus are exceptions to the department's Satisfactory Progress Policy.

Up to 4 credits of ENGR 321 (Engineering Cooperative Education) can be used as free elective credits toward the 180 total credits required for either degree.

Center for Career Services

The University's Center for Career Services, located in Mary Gates Hall, offers extensive services for students searching for career and job opportunities. Many local, national, and international companies send recruiters to campus each year to interview students in CompE, CS and other disciplines.

Research Opportunities

The department strongly supports the idea that undergraduates should have the opportunity to do research. Students interested in research opportunities should contact an undergraduate advisor or talk to a faculty member. In order to get credit for research, students should register for CSE 498 or 499. The department's research activities include most of the principal areas of Computer Science and Computer Engineering.

Scholarships

Scholarship opportunities are provided throughout the academic year. Scholarships are generally contributed by industry or federal agencies with awards based on academic performance, financial need, research contributions and student leadership. Awards can include monies, tuition and/or internships. Please refer to the department's web site for a listing and explanation of the scholarship opportunities: <http://www.cs.washington.edu/education/ugradscholars/scholarships.html>

Teaching Assistantships

Many departmental courses, from the introductory programming courses through the senior-level VLSI and graphics courses, use undergraduates as laboratory assistants and teaching assistants.

Honors Programs

Both degree programs offer an honors program. Students who qualify for Honors should request the appropriate form from an undergraduate advisor. Students must apply to graduate with honors at least one quarter in advance of graduation. Honors will be noted on the diploma.

Computer Science requirements for entry:

1. An overall GPA of at least 3.3.
2. A GPA of at least 3.5 for all Computer Science core courses and senior electives.
3. Affiliation with the College of Arts and Sciences.

Computer Engineering requirements for entry:

1. An overall GPA of at least 3.3.
2. A cumulative GPA of at least 3.5 for all Computer Science core courses and senior electives (you do not need a 3.5 in every course, the cumulative needs to be 3.5)
3. Affiliation with the College of Engineering.

To graduate with honors, a student must:

1. Maintain the above cumulative GPA.
2. Take nine credits of CSE 498H, and complete a Senior Thesis, which must satisfy the following requirements:
 - a. It should be a well-written document, clearly presented.
 - b. It should be work done independently by the student under the supervision of a faculty member. The work should be original and demonstrate creative thinking, as judged by the faculty supervisor and the Undergraduate Faculty Advisor.
 - c. Final approval of the thesis must be obtained from the faculty supervisor and the Undergraduate Faculty Advisor.



POLICIES AND PROCEDURES

Satisfactory Progress

Our program is designed to be pursued on a full-time basis. Students admitted to the Computer Science & Engineering undergraduate degree programs should plan to engage in full-time study. In addition, certain levels of academic performance are required to continue in the programs. The criteria and procedures below will be used in administering the Satisfactory Progress Policy of the department. These criteria and procedures are in addition to those of the University of Washington.

1. Students must complete three courses applicable toward their degree program each academic quarter in residence except summer quarter.
2. In exceptional circumstances students may petition to take fewer than the minimum requirements stated in item 1 above. See an advisor if you wish to petition.
3. The minimum acceptable grade for any CSE course is 2.0. A course may be taken a total of two times to satisfy this requirement. All other course requirements are subject to the University's minimum grade policy.
4. A student's overall grade point average must not fall below 2.0. If a course is taken more than once, all grades will be used in computing the grade point averages.
5. Graduation applications must be filed in accordance with University rules and regulations so that graduation takes place once all the requirements are satisfied.
6. Students who withdraw from the University without prior written approval of the Department, or are dropped for non-payment of fees, will forfeit their place in the program. Students must reapply for admission and, if re-admitted, must fulfill the requirements in effect at the time of re-admission.

Review of Progress

Each student's progress is reviewed at the conclusion of each quarter to ensure compliance with the criteria set forth above. In the event that a student fails to meet the criteria for satisfactory progress, a probation letter which indicates the reason(s) for the probationary status will be sent during the following quarter to the last address on record. In the absence of compelling extenuating circumstances, such a student will be placed on probation for failing to make satisfactory progress.

Students who fail to meet the criteria for satisfactory progress for two consecutive quarters in residence will be dismissed from the program. In addition, students who accumulate three probationary quarters will also be dismissed from the program. Such action will be communicated in writing to students at their last address on record and any current registrations or pre-registrations in Departmental courses will be canceled.

Academic Misconduct

Academic misconduct is any form of deception by which the student claims as his or her own work that which actually belongs to another, or otherwise gains unfair advantage over other students. This includes such actions as cheating on class assignments or examinations and plagiarism, as well as more severe offenses, such as theft or alteration of other academic materials for purposes of acquiring academic credit or enhancing grades.

All incidents of alleged academic misconduct will be reported to the Office of Academic Affairs. The Dean's Office will conduct a hearing unless the incident has been resolved to the satisfaction of the student and instructor involved.

In all considerations of academic misconduct, the department will carefully follow the letter and the spirit of the University's Student Conduct Code. This code has been separately printed in pamphlet form and is available to any interested student at <http://www.washington.edu/students/handbook/conduct.html>. The Student Conduct Code is made a part of this document by reference.

Another form of misconduct is the misuse of computing facilities, for example, obtaining improper access to another student's account. Before students obtain accounts on departmental computers they sign an agreement stating that they will not misuse their accounts. Violations of this agreement may result in the loss of computer accounts and/or being dropped from the major.

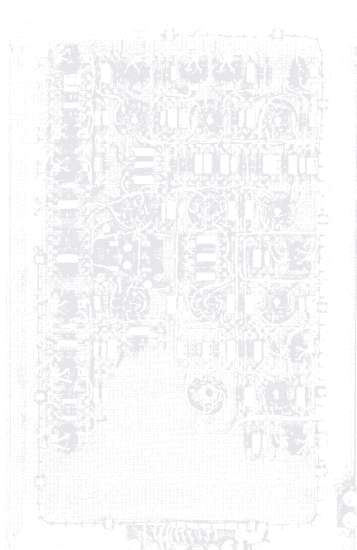
Each student who accepts admission into the Computer Science & Engineering degree programs agrees to be bound by the provisions of this section.

Graduation

It is the responsibility of Computer Science & Engineering students to ensure that they meet all of the degree requirements. Assistance in the program planning is available from an undergraduate advisor. Application for graduation should be made with an advisor at least two quarters prior to the quarter students wish to graduate.

Applicability

The requirements contained in this Handbook apply to all students admitted in or after Autumn 2000 and until such time as a new Handbook or amendment is issued. Students admitted while this Handbook is current will not be subject to additional requirements that might be issued in the future, except those found necessary to comply with accreditation requirements.



Course Listings

The coursework offered in the Computer Science & Engineering curriculum can be divided into three categories: introductory, majors-only, and non-majors courses.

Introductory courses are open to all students and consist of our introductory programming courses. These serve as prerequisites for all other courses in the department.

Majors-only courses are open to computer science and computer engineering majors only. They give students in-depth experience in the areas necessary to the study of their discipline. They include common core courses, 400-level advanced courses which students choose based upon the option they are pursuing (computer science, computer engineering hardware or computer engineering software), and capstone design courses.

Courses for non-majors are intended to give a technical introduction to fundamental topics in computer science to students not majoring in the department, but who are likely to use computers as tools in their own disciplines. Prospective computer science or computer engineering majors should not take these courses. Instead, they should wait until they are admitted to the major and take the majors courses listed in the majors-only course section.

Introductory Computer Science and Engineering Courses

CSE 100 FLUENCY IN INFORMATION TECHNOLOGY (5)

Introduces skills, concepts, and capabilities necessary to effectively use information technology. Includes logical reasoning, managing complexity, operation of computers and networks, and contemporary applications such as effective Web searching and database manipulation, ethical aspects, and social impacts of information technology. Not available for credit to students who have completed CSE 142 or ENGR 142. Offered: jointly with INFO 100.

CSE 142 COMPUTER PROGRAMMING I (4)

Basic programming-in-the-small abilities and concepts. Highlights include procedural and functional abstraction with simple built-in data type manipulation. Basic abilities of writing, executing and debugging programs. Not available for credit to students who have completed CSE 210 or ENGR 141.

CSE 143 COMPUTER PROGRAMMING II (5)

Continuation of CSE 142. Concepts of modularity and encapsulation, focusing on modules and abstract data types. Covers some basic data structures. Not available for credit to students who have completed CSE 211. Prerequisite: CSE 142.

Computer Science and Engineering Courses for Majors

CSE 321 DISCRETE STRUCTURES (4)

Fundamentals of set theory, graph theory, enumeration, and algebraic structures, with applications in computing. Prerequisite: CSE 143 and either MATH 123, MATH 129, or MATH 136.

CSE 322 INTRODUCTION TO FORMAL MODELS IN COMPUTER SCIENCE (3)

Finite automata and regular expressions; context free grammars and pushdown automata; nondeterminism; Turing machines and the halting problem. Emphasis on understanding models and their applications and on rigorous use of basic techniques of analysis. Induction proofs, simulation, diagonalization and reduction arguments. Prerequisites: CSE 321.

CSE 326 DATA STRUCTURES (4)

Data types, abstract data types, and data structures. Efficiency of algorithms. Sequential and linked implementation of lists. Binary tree representations and traversals. Searching: dictionaries, priority queues, hashing. Directed graphs, depth-first algorithms. Garbage collection. Dynamic storage allocation. Internal and external sorting. Prerequisite: CSE 321. No credit if CSE 373, CSE 374, or EE 374 has been taken.

CSE 341 PROGRAMMING LANGUAGES (4)

Designed to make the student reasonably fluent in several radically different languages, such as LISP, Prolog, and Smalltalk. Prerequisite: CSE 143. No credit if CSE 413 has been taken.

CSE 370 INTRODUCTION TO DIGITAL DESIGN (4)

Introductory course in digital logic and its specification and simulation. Boolean algebra, combinational circuits including arithmetic circuits and regular structures, sequential circuits including finite-state-machines, use of programmable logic devices. Simulation and high-level specification techniques are emphasized.

CSE 378 MACHINE ORGANIZATION AND ASSEMBLY LANGUAGE (4)

Difference and similarities in machine organization; central processors; fundamentals of machine language and addressing; assembly language programming, including macros; operating system interfaces. Prerequisite: CSE 143, CSE 378. No credit if CSE 410 has been taken.

CSE 401 INTRODUCTION TO COMPILER CONSTRUCTION (3)

Fundamentals of compilers and interpreters; symbol tables, lexical analysis, syntax analysis, semantic analysis, code generation, and optimization for general purpose programming languages. Prerequisites: CSE 322, CSE 326, CSE 341, and CSE 378. No credit if CSE 413 has been taken.

CSE 403 SOFTWARE ENGINEERING (4)

This course presents the 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 326; CSE 341. Recommended: project experience in an academic or work setting.

CSE 421 INTRODUCTION TO ALGORITHMS (3)

Techniques for design of efficient algorithms. Methods for showing lower bounds on computational complexity. Particular algorithms for sorting, searching, set-manipulation, arithmetic, graph problems, pattern-matching, etc. Prerequisites: CSE 322 and CSE 326.

CSE 431 INTRODUCTION TO THEORY OF COMPUTATION (3)

Models of computation, computable and non-computable functions, space and time complexity, tractable and intractable functions. Prerequisite: CSE 322.

CSE 444 INTRODUCTION TO DATABASE SYSTEMS (3)

Fundamental concepts, system organization and implementation of database systems will be covered. Topics include the relational, hierarchical and network data models, file organizations and data structures, query languages, query optimization, database design, concurrency control, security, and issues involving distributed database systems. Prerequisite: CSE 326.

CSE 451 INTRODUCTION TO OPERATING SYSTEMS (4)

Principles of operating systems. Process management, memory management, auxiliary storage management, resource allocation. Prerequisites: CSE 326 and CSE 378. No credit if CSE 410 or EE 474 has been taken.

CSE 455 COMPUTER VISION (4)

Introduction to image analysis and interpreting the 3D world from image data. Topics may include feature detection, image segmentation, motion estimation, image mosaics, 3D-shape reconstruction, object recognition, and image retrieval. Prerequisite: CSE 326.

CSE 457 COMPUTER GRAPHICS (4)

Techniques of computer image synthesis, including both hardware and software. Line drawing and color raster graphics; geometric transformations, hidden surface and smooth shading algorithms. Prerequisite: CSE 326.

CSE 458 COMPUTER ANIMATION (5)

Introduction to basic principles of computer generated animation. Focus on the modeling and lighting of animated characters. Students from Art, CSE, and Music team up on projects to be built on commercially-available modeling and lighting packages. Prerequisite: CSE 457 or ART 380 or MUSIC 403 and permission of instructor.

CSE 461 INTRODUCTION TO COMPUTER-COMMUNICATION NETWORKS (4)

Computer network architectures, protocol layers. Transmission media, encoding systems, error detection, multiplexing, switching. Data link, multiple access channel protocols. Methods for network routing, congestion control, flow control. End-to-end transport services, protocols. Network security, privacy. Applications including electronic mail, virtual terminals, distributed operating systems. Prerequisite: CSE 143; either MATH 390/STAT 390, STAT 391, IND E 315, or CSE 321. Offered: jointly with E E 461.

CSE 466 SOFTWARE FOR EMBEDDED SYSTEMS (4)

Software issues in the design of embedded systems. Microcontroller architectures and peripherals, embedded operating systems and device drivers, compilers and debuggers, timer and interrupt systems, interfacing of devices, communications and networking. Emphasis on practical application of development platforms. Prerequisites: CSE 326, CSE 370 and CSE 378.

CSE 467 ADVANCED DIGITAL DESIGN (3)

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 326 and CSE 370.

CSE 468 VERY LARGE SCALE INTEGRATION (5)

Introduction to CMOS technology and circuit design; implementation of combinational and sequential logic; VLSI design methodologies; CAD tools for layout, simulation and validation. Students

CSE 471 COMPUTER DESIGN AND ORGANIZATION (4)

CPU instruction addressing models, CPU structure and functions, computer arithmetic and logic unit, register transfer level design, hardware and microprogram control, memory hierarchy design and organization, I/O and system components interconnection. Laboratory project involves design and simulation of an instruction set processor. Prerequisite: CSE 370 and CSE 378.

CSE/LING 472 INTRODUCTION TO COMPUTATIONAL LINGUISTICS (3)

Introduction to computer applications of linguistic theory, including syntactic processing, semantic, and pragmatic interpretation and natural language generation. Prerequisite: Ling 471, or permission of instructor.

CSE 473 INTRODUCTION TO ARTIFICIAL INTELLIGENCE (3)

Principal ideas and developments in artificial intelligence: theorem proving, problem-solving methods; representation of knowledge, natural language analysis and synthesis, programming languages for artificial intelligence. Prerequisites: CSE 326, CSE 341 recommended. No credit if CSE 415 has been taken.

CSE 476 EMBEDDED SYSTEM DESIGN (5)

System building course to provide students with a complete experience in embedded system design. Students will design, simulate, construct, debug, and document a substantial project of their choosing. Lectures will focus on case studies and emerging components and platforms. Prerequisites: CSE451 and CSE466.

CSE 477 DIGITAL SYSTEM DESIGN (5)

Students use the laboratory to design, simulate, construct, and debug a substantial project that includes hardware, software, and communication components. Lectures focus on use of embedded processors in digital system design and interfacing techniques. Writing and debugging of real-time reactive software emphasized. Prerequisites: CSE 378 and CSE 467.

CSE 481 CAPSTONE SOFTWARE DESIGN (5)

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: Senior standing in CSE or permission of the instructor.

CSE 490 SPECIAL TOPICS IN COMPUTER SCIENCE & ENGINEERING (1-5)

Lectures and discussions on topics of current interest in computer science and engineering not covered by other CSE undergraduate courses.

CSE 498, 498H SENIOR PROJECT (1-6) HONORS (9)

Consists of a report (and perhaps demonstration) describing a development, survey, or small research project completed by the student in an area of specialization. Objectives are: (1) applying and integrating the classroom material from several courses, (2) introducing the professional literature, (3) gaining experience in writing a technical document, and (4) enhancing employability through the evidence of independent work. The project may cover an area in computer science and engineering or an application to another field. The work normally extends over more than one quarter. Prerequisite: Senior standing and permission of instructor.

CSE 499 READING AND RESEARCH (1-24)

Available in special situations for advanced CSE majors to do reading and research in the field. Subject to approval of the undergraduate advisor and a Computer Science and Engineering faculty member. Usable as a free elective, but not in place of a core course or Computer Engineering elective. Prerequisite: Senior standing and permission of instructor. Credit/no credit.

Computer Science and Engineering Courses for Non-Majors

CSE 100 FLUENCY IN INFORMATION TECHNOLOGY (5)

Introduces the skills, concepts, and capabilities necessary to effectively use information technology. Includes logical reasoning, managing complexity through the operation of computers and networks, and contemporary applications such as email and word processing. Not available for credit to students who have completed CSE 142 or ENGR 142. Offered jointly with IMT 100.

CSE 135 COMPUTATIONAL TECHNIQUES (4)

Principles of computation and techniques for using the computer as a tool in science and engineering. Basic programming techniques (conditionals, iteration, procedural, and data abstraction). Introduction to simulation, visualization, and symbolic mathematics. Solving engineering and science problems using programmable tools rather than conventional programming languages. Prerequisites: Math 124. Note: this course is not a requirement for either the Computer Science or the Computer Engineering degree.

CSE 373 DATA STRUCTURES AND ALGORITHMS (3)

Fundamental algorithms and data structures for their implementation. Techniques for solving problems by programming. Linked lists, stacks, queues, directed graphs. Trees: representations and traversals. Searching (hashing, binary search trees, multiway trees). Garbage collection, memory management. Internal and external sorting. Prerequisite: CSE 143. No credit if CSE 326, CSE 374, or EE 374 has been taken.

CSE 410 COMPUTER SYSTEMS (3)

Structure and components of hardware and software systems. Machine organization, including central processor and input-output architectures; assembly language programming; operating systems, including process, storage, and file management. Prerequisite: CSE 373. No credit if CSE 378 or CSE 451 has been taken.

CSE 413 PROGRAMMING LANGUAGES AND THEIR IMPLEMENTATION (3)

Concepts and implementation strategies for Algol-class languages, including Pascal, Modula, Algol 60, Ada. Compilers for Algol-class languages. Languages with late binding times, including LISP, APL, Smalltalk. Prerequisite CSE 373. No credit if CSE 341 or CSE 401 has been taken.

CSE 415 INTRODUCTION TO ARTIFICIAL INTELLIGENCE (5)

Principles and programming techniques of artificial intelligence: LISP, symbol manipulation knowledge representation, logical and probabilistic reasoning, learning, language understanding, vision, expert systems, and social issues. Prerequisites: CSE 373, or CSE 143 and permission of instructor. No credit if CSE 473 has been taken.

CSE 417 ALGORITHMS AND COMPUTATIONAL COMPLEXITY (3)

Design and analysis of algorithms and data structures. Efficient algorithms for manipulating graphs and strings. Fast Fourier Transform. Models of computation, including Turing machines. Time and space complexity. NP-complete problems and undecidable problems. Prerequisite: CSE 373.



Computer Science and Engineering Faculty

Richard Anderson, Professor (1986) Ph.D., 1985, Stanford University.
Educational technology, algorithms..

Tom Anderson, Associate Professor (1997) Ph.D., 1991, University of Washington.
Internetworking, local and wide area distributed systems, operation systems, computer architecture.

Larry Arnstein, Research Assistant Professor (1999) Ph.D., 1997, Carnegie Mellon University.
Ubiquitous computing, bioinformatics, embedded systems design.

Jean-Loup Baer, Professor, Adjunct Professor of Electrical Engineering (1969) Ph.D., 1968, University of California at Los Angeles.
Computer architecture and performance evaluation.

Paul Beame, Professor (1987) Ph.D., 1987, University of Toronto.
Computational complexity, proof complexity..

Brian Bershad, Associate Professor (1993) Ph.D., 1990, University of Washington.
Operating systems, architecture, distributed systems, parallel systems.

Alan Borning, Professor, Adjunct Professor in the Information School, Member of the Interdisciplinary Ph.D. program in Urban Design and Planning (1980) Ph.D., 1979, Stanford University.
Human-computer Interaction, integrated land use, transportation and environmental modeling, and constraint-based languages and systems.

Gaetano Borriello, Professor, (1988) Ph.D., 1988, University of California at Berkeley.
Invisible and ubiquitous computing, multi-modal user interfaces, software architectures for embedded and networked systems..

Craig Chambers, Associate Professor (1991) Ph.D., 1992, Stanford University.
Programming language design, optimizing compilation, object-oriented systems.

Brian Curless, Assistant Professor (1998) Ph.D., 1997, Stanford University.
Computer graphics, 3D photography.

Martin Dickey, Senior Lecturer (1996) Ph.D., 1992, Arizona State University.
Computational linguistics, computer science education.

Chris Diorio, Associate Professor, Adjunct Associate Professor of Electrical Engineering (1997) Ph.D., 1997, California Institute of Technology..
Silicon learning chips, neural networks and learning algorithms, implantable microcontrollers for studying neural signaling in intact behaving animals.

Pedro Domingos, Assistant Professor (1999) Ph.D., 1997, University of California at Irvine.
Artificial intelligence, machine learning, data mining.

Emer Dooley, Lecturer (2000) Ph.D., 2000, University of Washington
Entrepreneurship and strategic management in high-tech industries, competitive dynamics and sustainable competitive advantage.

Carl Ebeling, Professor, Adjunct Professor of Electrical Engineering (1986) Ph.D., 1986, Carnegie-Mellon University.
VLSI architectures, configureable computing, computer-aided design.

Susan Eggers, Professor (1989) Ph.D., 1989, University of California at Berkeley.
Multithreaded and domain-specific architectures and program behavior, back-end compiler optimizations, dynamic (staged) optimization.

Oren Etzioni, Associate Professor and Career Development Scholar (1991) Ph.D., 1990, Carnegie-Mellon University.
Artificial intelligence: machine learning, integrated architectures, planning.

Dieter Fox, Assistant Professor, Assistant Professor of Electrical Engineering (2000) Ph.D., 1998, University of Bonn, Germany.

Artificial intelligence and mobile robotics, probabilistic state estimation, particle filters.

Steven Gribble, Assistant Professor (2000) Ph.D., University of California at Berkeley.

Operating systems, distributed systems, cluster computing, Internet infrastructure and scalable services, and mobile computing.

Venkatesan Guruswami, Assistant Professor (2002) Ph.D., 2001, Massachusetts Institute of Technology. Coding Theory, approximation algorithms, complexity theory.

Alon Halevy, Associate Professor (1998). Ph.D., 1993, Stanford University.

Database systems, artificial intelligence, data integration, peer-based data management, knowledge representation.

Anna R. Karlin, Professor (1996) Ph.D., 1987, Stanford University.

Online algorithms, probabilistic algorithms and probabilistic analysis

Henry Kautz, Associate Professor (2000) Ph.D., 1998 University of Rochester.

Artificial intelligence, planning, decision theory, verification, satisfiability testing, ubiquitous computing, AI aids for cognitive disabilities..

Richard Ladner, Professor (1971). Ph.D., 1971, University of California at Berkeley.

Design and analysis of algorithms, data compression, network algorithms, cache performance, computational complexity, computers to aid disabled persons.

Edward D. Lazowska, Bill & Melinda Gates Chair (1977). Ph.D., 1977, University of Toronto.

Computer systems: modeling and analysis, design and implementation, distributed and parallel systems.

Henry M. Levy, Microsoft Professor (1983). M.S., 1981, University of Washington.

Operating systems, distributed and parallel systems, WWW and web performance, computer architecture, multithreaded processors.

Barbara Mones, Senior Lecturer, Creative Director for Digital Animation, (1999). M.F.A., 1979, Rhode Island School of Design.

Computer graphics, character animation.

David Notkin, Boeing Professor and Chair (1984). Ph.D., 1984, Carnegie-Mellon University.

Software engineering, software evolution, software dependability.

Mark Oskin, Assistant Professor (2001) Ph.D., 2001 University of California at Davis.

Computer architecture, intelligent memory systems, dataflow, quantum computing.

Hal Perkins, Senior Lecturer (1998). M.S., 1982, Cornell University.

Computer science education, programming languages and compilers.

Zoran Popovic, Assistant Professor (1999). Ph.D., 1999, Carnegie-Mellon University.

Computer graphics, character animation, physically based modeling simulation.

Rajesh Rao, Assistant Professor (2000). Ph.D., 1998, University of Rochester.

Neural computing, machine vision and learning, robotics, computational neuroscience, brain-computer interfaces.

Walter L. Ruzzo, Professor (1977) Ph.D., 1978, University of California at Berkeley.

Computational biology.

David Salesin, Professor (1992). Ph.D., 1991, Stanford University.

Computer graphics.

Steve Seitz, Assistant Professor (2000). Ph.D., 1997, University of Wisconsin.
Computer vision, computer graphics.

Linda Shapiro, Professor, Professor of Electrical Engineering, Adjunct Professor of Medical Education and Biomedical Informatics, (1986). Ph.D., 1974, University of Iowa.

Computer vision, multimedia information systems, medical informatics, pattern recognition and learning, robotics.

Lawrence Snyder, Professor (1983). Ph.D., 1973, Carnegie-Mellon University.
Parallel computation, especially hardware, languages and algorithmic issues, and computer fluency for beginners.

Dan Suciu, Associate Professor (2000). Ph.D., 1995, University of Pennsylvania.
Database systems, database theory, XML data management, stream XML data processing, peer-based data management.

Steven L. Tanimoto, Professor, Adjunct Professor of Electrical Engineering (1977). Ph.D., 1975, Princeton University.
Visual languages, image analysis, computer graphics, artificial intelligence, educational technology.

Martin Tompa, Professor (1978). Ph.D., 1978, University of Toronto.
Computational molecular biology, computational complexity.

Dan Weld, Professor (1988). Ph.D., 1988, Massachusetts Institute of Technology.
Artificial Intelligence, intelligent user interfaces, software agents, planning, database and data integration.

David Wetherall, Assistant Professor (1999). Ph.D., 1998, Massachusetts Institute of Technology.
Networks, distributed systems.

John Zahorjan, Professor (1980). Ph.D., 1980, University of Toronto.
Computer systems; system security; performance analysis; multimedia systems.