Computer Science and Engineering (CSE)

OFFICES:
Undergraduate Affairs 3402
Graduate Affairs 3402
Applied Physics and Mathematics Building, Muir College
http://www.cs.ucsd.edu

Professors

Donald W. Anderson, Ph.D., *Emeritus*
Richard K. Belew, Ph.D.
Francine D. Berman, Ph.D.
Kenneth L. Bowles, Ph.D., *Emeritus*
Walter A. Burkhard, Ph.D.
J. Lawrence Carter, Ph.D.
Chung-Kuan Cheng, Ph.D.
Andrew Chien, Ph.D.
Garrison W. Cottrell, Ph.D.
Jeanne Ferrante, Ph.D.
Joseph A. Goguen, Ph.D.
Fan Chung Graham, Ph.D.
Ronald Graham, Ph.D.
William E. Howden, Ph.D.
Russell Impagliazzo, Ph.D.
T. C. Hu, Ph.D.

Professors

Donald W. Anderson, Ph.D., *Emeritus*
Scott B. Baden, Ph.D.
Mihir Bellare, Ph.D.
Francine D. Berman, Ph.D.
Kenneth L. Bowles, Ph.D., *Emeritus*
Walter A. Burkhard, Ph.D.
J. Lawrence Carter, Ph.D.
Chung-Kuan Cheng, Ph.D.
Andrew Chien, Ph.D.
Garrison W. Cottrell, Ph.D.
Jeanne Ferrante, Ph.D.
Joseph A. Goguen, Ph.D.
Fan Chung Graham, Ph.D.
Ronald L. Graham, Ph.D.
Rajesh Gupta, Ph.D.
William E. Howden, Ph.D.
T. C. Hu, Ph.D.
Russell Impagliazzo, Ph.D.
Ramesh C. Jain, Ph.D., *Emeritus*
Andrew B. Kahng, Ph.D.
Sidney Karin, Ph.D. *(In-Residence)*
David Kriegman, Ph.D.
Keith Marzullo, Ph.D.
Alex Orailoglu, Ph.D.
Alon Orlitsky, Ph.D.
Joseph C. Pasquale, Ph.D.
Ramamohan Paturi, Ph.D.
Pavel A. Pevzner, Ph.D.
Venkat P. Rangan, Ph.D.
Walter J. Savitch, Ph.D.
Larry L. Smarr, Ph.D.
George Varghese, Ph.D.
Victor D. Vianu, Ph.D.
S. Gill Williamson, Ph.D.

Associate Professors

Bradley G. Calder, Ph.D.
Charles P. Elkan, Ph.D.
William G. Griswold, Ph.D.
Dean M. Tullsen, Ph.D.
Alexander Vardy, Ph.D.

Assistant Professors

Serge J. Belongie, Ph.D.
Sanjoy Dasgupta, Ph.D.
Alin Deutsch, Ph.D.
Henrick Wann Jensen, Ph.D.
Ingolf Krueger, Ph.D. *(In-Residence)*
Daniele Micciancio, Ph.D.
Yannis Papakonstantinou, Ph.D.
Stefan Savage, Ph.D.
Alex Snoeren, Ph.D.
Geoffrey Voelker, Ph.D.
Bennet S. Yee, Ph.D.

Adjunct Faculty

Michael J. Bailey, Ph.D.
Samuel R. Buss, Ph.D.
Henri Casanova, Ph.D.
Walter H. Ku, Ph.D.
Bertram Ludaescher, Ph.D.
Reagan Moore, Ph.D.
J. Benjamin Rosen, Ph.D.
George Polyzos, Ph.D.
Jeffrey B. Remmel, Ph.D.
Terrence J. Sejnowski, Ph.D.

Research Faculty

Philip Papadopoulos, Ph.D., *Associate Research Scientist*

Academic Coordinator

Paul Kube, Ph.D., *Lecturer with Security of Employment*

The Undergraduate Programs

The Department of Computer Science and Engineering offers computer science and computer engineering curricula leading to the degrees in B.S. in Computer Science, B.S. in Computer Engineering, B.A. in Computer Science, and B.S. in Computer Science with a specialization in Bioinformatics. The courses of study prepare students for graduate study in these fields as well as immediate employment. The B.A. degree is intended to provide a more flexible program of study allowing significant studies beyond computer science and engineering.

These degrees are four-year endeavors. Students in the B.S. programs need to enroll in no more than sixteen units per quarter during their junior and senior years to meet their major requirements. The B.A. program has fewer major requirements. In addition, each student must satisfy general-education course requirements determined by the student's college.

B.S. Computer Science Program

The lower-division B.S. computer science program is designed to provide a strong foundation in mathematics, physics, electrical engineering, programming methodology and skills, and computer organization. Upper-division core courses deal with the theory and design of algorithms, hardware, and software. Students can gain additional breadth and/or depth in computer science and engineering by an appropriate selection of technical electives.

Students should have sufficient background in high school mathematics so that they can take freshman calculus in their first quarter. Courses in high school physics and computer programming, although helpful, are not required for admission to the program.

The department requires a total of 134 units for the B.S. computer science program (not including the general-education requirements). There are three varieties of requirements: lower-division, upper-division, and technical electives.

1. LOWER-DIVISION REQUIREMENTS

Students are expected to complete the following seventy units by the end of their sophomore year.

Computer Science and Engineering: CSE 8B or 11, 12, 20 or Math. 15A, CSE 21 or Math. 15B, and CSE 30; twenty units.

Return To Contents

Note: Students without any programming experience are advised to take CSE 8A and then CSE 8B, instead of CSE 11. CSE 11 is a faster paced version of CSE 8A and CSE 8B, and requires experience in programming with a compiled language.

Mathematics: Math. 20A-B-C-D and 20F; twenty units.

Physics: Phys. 2A-B-C; 12 units. Math. 20A is a prerequisite for Phys. 2A. Students whose performance on the Department of Mathematics placement test permits them to start with Math. 20B or a higher course may take Phys. 2A in the fall quarter of the freshman year; all others will take Phys. 2A in the winter quarter of the freshman year. Students who received high grades in both calculus and Physics in high school may substitute the major's sequence, Phys. 4A-B-C for Phys. 2A-B-C.

Physics Lab: Phys. 2BL or 2CL or 2DL; two units. The lab course should be taken concurrently with the Phys. 2 or Phys. 4 sequence.

Introduction to Electrical Engineering: ECE 53A-B; eight units. ECE 53A-B are courses that give a comprehensive introduction to electrical engineering.

Probability and Statistics: Math.183; four units.

Science/Mathematics Elective: Students are required to take one of the following four-unit science/mathematics courses: Phys. 2D, Math. 20E(2F), Chem. 6A, BILD 1, 10, 12, 14, 30.

2. UPPER-DIVISION REQUIREMENTS

All B.S. computer science students are required to take CSE 100 or Math. 176, CSE 101 or Math. 188, CSE 105 or Math. 166, CSE 120, 130, 131A-B, 140, 140L, 141 and 141L; forty units.

Students are expected to complete almost all of these courses by the end of their junior year. If students want to accelerate their program, they should consider taking CSE 100 or Math. 176, CSE 105, and/or CSE 140 and 140L in the sophomore year.

3. TECHNICAL ELECTIVES

B.S. computer science students are required to take six technical electives for a total of twenty-four units. Four electives must be computer science and engineering upper-division or graduate courses.

The remaining two technical electives can be chosen from the wider set of courses that includes computer science and engineering upper-division courses, graduate courses, and other electives as listed under the section titled **Electives**. Other restrictions in the selection of technical electives are also given in the section **Electives**.

4. B.S. COMPUTER SCIENCE, SAMPLE PROGRAM

FALL	WINTER	SPRING
FRESHMAN YEAR		
CSE 8A or CSE 11	CSE 12	CSE 20 or Math. 15A
Math. 20A	Math. 20B	Math. 20C
GE	Phys. 2A	Phys. 2B
	GE	GE
SOPHOMORE YEAR		
CSE 21 or Math. 15B	Sci/Math. Elec.	CSE 30
Math. 20D	Math. 20F	Math. 183
GE	ECE 53A	Phys. 2BL or 2CL or 2DL
GE	Phys. 2C	ECE 53B
JUNIOR YEAR		
CSE 100 or Math. 176	CSE 101 or Math. 188	CSE 120
CSE 140	CSE 141	CSE 130
CSE 140L	CSE 141L	CSE Tech. Elec.
CSE 105 or Math. 166	CSE Tech. Elec.	GE
GE	GE	
SENIOR YEAR		
CSE 131A	CSE 131B	CSE Tech. Elec.
CSE Tech. Elec.	Tech. Elec.	Tech. Elec.
GE	GE	GE

B.S. Computer Engineering Program

(Curriculum is the same in both the CSE and ECE departments.)

The B.S. computer engineering program is jointly administered by the Departments of Computer Science and Engineering and Electrical and Computer Engineering. Students wishing to take the computer engineering program must be admitted to one of the departments.

The lower-division computer engineering program is designed to provide a strong foundation in mathematics, physics, electrical engineering, programming methodology and skills, and computer organization. Upper-division core courses deal with the theory and design of algorithms, hardware and software, as well as electronic systems. Students can gain additional breadth and/or depth in computer science and engineering by an appropriate selection of technical electives.

Students should have sufficient background in high school mathematics so that they can take freshman calculus in their first quarter. Courses in high school physics and computer programming, although helpful, are not required for admission to the program.

B.S. computer engineering program requires a total of 146 units (not including the general-education requirements). There are three varieties of requirements: lower-division, upper-division, and technical electives.

1. LOWER-DIVISION REQUIREMENTS

Students are expected to complete the following seventy units by the end of their sophomore year.

Computer Science and Engineering: CSE 8B or 11, 12, 20 or Math. 15A, CSE 21 or Math. 15B, and CSE 30; twenty units.

Note: Students without any programming experience are advised to take CSE 8A and then CSE 8B, instead of CSE 11. CSE 11 is a faster paced version of CSE 8A and CSE 8B, and requires experience in programming with a compiled language.

Mathematics: Math. 20A-B-C-D and 20F; twenty units.

Physics: Phys. 2A-B-C-D; sixteen units. Math. 20A is a prerequisite for Phys. 2A. Students whose performance on the Department of Mathematics placement test permits them to start with Math. 20B or a higher course may take Phys. 2A in the fall quarter of the freshman year; all others will take Phys. 2A in the winter quarter of the freshman year. Students who received high grades in both calculus and physics in high school may substitute the major's sequence, Phys. 4A-B-C-D for Phys. 2A-B-C-D.

Physics Lab: Phys. 2BL or 2CL or 2DL; two units. The lab courses should be taken concurrently with the Phys. 2 or Phys. 4 sequence.

Introduction to Electrical Engineering: ECE 53A-B; eight units. ECE 53A-B are courses that give a comprehensive introduction to electrical engineering.

Probability and Statistics: ECE 109; four units. This course can be taken in the sophomore year.

2. UPPER-DIVISION REQUIREMENTS

All B.S. computer engineering students are required to take CSE 100 or Math. 176, CSE 101 or Math. 188, CSE 105 or Math. 166, CSE 120, 131A-B, 140, 140L, 141, and 141L; thirty-six units.

In addition, all B.S. computer engineering students have to fulfill the following upper-division ECE requirements.

Linear Systems: ECE 101, ECE 171A or 161A; eight units. The department recommends that these courses be taken in the junior year.

Electronic Circuits and Systems: ECE 102, ECE 108; eight units. The department recommends that these courses be taken in the junior year.

If students want to accelerate their program, they should consider taking CSE 100 or Math. 176, CSE 105 or Math. 166, and/or CSE 140 and 140L in the sophomore year.

3. TECHNICAL ELECTIVES

All B.S. computer engineering students are required to take six technical electives for a total of twenty-four units. One of these courses must be either ECE 111 or ECE 118. Of the remaining five courses, four must be computer science and engineering or electrical and computer engineering upper-division or graduate courses.

The remaining course can be any computer science and engineering or electrical and computer engineering upper-division or graduate course, or any other course listed under the section titled **Electives**. Other restrictions in the selection of technical electives are also given in the section **Electives**.

4. B.S. COMPUTER ENGINEERING, SAMPLE PROGRAM

FALL	WINTER	SPRING
FRESHMAN YEAR		
CSE 8A or CSE 11 Math. 20A GE	CSE 12 Math. 20B Phys. 2A GE	CSE 20 or Math. 15A Math. 20C Phys. 2B GE
SOPHOMORE YEAR		
CSE 21 or Math. 15B Math. 20D Phys. 2C GE	Math. 20F ECE 53A ECE 109 GE	CSE 30 ECE 53B Phys. 2D Phys. 2BL or 2CL or 2DL
JUNIOR YEAR		
CSE 100 or Math. 176 CSE 140 CSE 140L ECE 102 GE	CSE 101 or Math. 188 CSE 141 CSE 141L ECE 108 GE	CSE 105 or Math. 166 CSE 120 (Req. Tech. Elec.- ECE 111 or ECE 118) GE
SENIOR YEAR		
CSE 131A ECE 101 CSE/ECE Tech. Elec. GE.	CSE 131B CSE/ECE Tech. Elec. CSE/ECE Tech. Elec GE	ECE 171A or 161A CSE/ECE Tech. Elec. Tech. Elec. GE

B.S. Computer Science with a specialization in Bioinformatics

The explosion in biological knowledge spawned by the various genome projects has created entirely new fields and industries, and a need for trained computational biologists who are familiar with biology, mathematics, and computer sciences. The Computer Science and Engineering Department offers rigorous, interdisciplinary training in the new and rapidly evolving field of bioinformatics. Bioinformatics refers to advanced computational and experimental methods that model the flow of information (genetic, metabolic, and regulatory) in living systems to provide an integrated understanding of the system properties of organisms. This interdisciplinary major will be offered by three other programs (Division of Biology, Department of Chemistry and Biochemistry, and Department of Bioengineering). The computer science and engineering requirements comprise of 152 units to be taken from the divisions of physical sciences, biology, and engineering.

1. LOWER-DIVISION REQUIREMENTS

Lower-division requirements, 64 units: Students are expected to complete all lower-division requirements by the end of their sophomore year.

1. Math. 20A, 20B, 20C, 20F, (16 units)
2. Chemistry 6A, 6B, 6C, and one lab (15 units)
3. BILD 1, BILD 2, and BILD 94 (9 units)
4. CSE 11, CSE 12, and CSE 21 or Math. 15B (12 units)
5. Physics 2A, 2B, 2C (12 units)

2. UPPER-DIVISION REQUIREMENTS

Upper-division requirements, 88 units (includes five CSE technical electives)

1. CSE 100 or Math. 176 (Data Structures), (4 units)
2. CSE 101 or Math. 188 (Algorithms), (4 units)
3. Chemistry 140A–140B (Organic Chemistry), (8 units)
4. Chemistry 114B (Biochemical Energetics and Metabolism) or BIBC 102 (Structural and Metabolic Biochemistry) (4 units)
5. BIBC 103 (Biochemical Techniques), (4 units)
6. BICD 100 (Genetics), (4 units)
7. BIMM 100 (Molecular Biology) or Chemistry 114D (Molecular and Cellular Biochemistry), (4 units)
8. BIMM 101 (Recombinant DNA Lab), (4 units)
9. BICD 110 (Cell Biology), (4 units)
10. BIBC 110 (Physical Biochemistry) or Chemistry 127 (Physical Chemistry), (4 units)
11. Five additional CSE upper-division electives (electives 1, 2, 3, 4, and 5).

    At least one course from each of the three groups for a total of five electives:

    Group I: CSE 30, 111, 131A, 131B, 134A

    Group II: CSE 105, 150, 151, Math. 184A

    Group III: CSE 132A, 132B, 133

The bioinformatics series comprised of the following six courses, 24 units:

12. CSE 181 or BIMM 181 or BENG 181 (Molecular Sequence Analysis), (4 units)
13. CSE 182 or BIMM 182 or BENG 182 or Chem. 182 (Biological Databases), (4 units)
14. BENG 183 (Applied Genomic Technologies), (4 units)
15. CSE 184 or BIMM 184 or BENG 184 (Computational Molecular Biology), (4 units)
16. BIMM 185 (Bioinformatics lab), (4 units)
17. Math. 186 (Probability and Statistics), (4 units)

3. B.S. COMPUTER SCIENCE WITH A SPECIALIZATION IN BIOFORMATICS, SAMPLE PROGRAM

FALL	WINTER	SPRING
FRESHMAN YEAR		
CSE 8A+8B or 11[1] Math. 20A Chem. 6A GE 1	CSE 12 Math. 20B Chem. 6B Chem. 6BL GE 2	BILD 94[2] Math. 20C Chem. 6C BILD 1 GE 3
SOPHOMORE YEAR		
CSE 21 or Math. 15B BILD 2 Phys. 2A GE 4	Math. 20F Phys. 2B Chem. 140A GE 5	Chem. 140B BIBC 103 Phys. 2C GE 6

JUNIOR YEAR		
CSE 100 or	CSE 101 or	CSE 181[4]
Math. 176[3]	Math. 188	BIMM 101
or		
BICD 100	Math. 186	Chem.
112B		
GE 7	BIBC 102 or	BICD 110
GE 8	Chem. 114B	BIBC 110 or
	BIMM 100 or	Chem.
127		
	Chem. 114D	
SENIOR YEAR		
CSE 182	CSE 184	BIMM 185
BENG 183	Elec. 2	Elec. 4
Elec. 1[5]	Elec. 3	Elec. 5
GE 9	GE 10	GE 11

[1] Students may take the slower paced version, CSE 8A + CSE 8B, instead of CSE 11.

[2] BILD 94 (1 unit seminar) is recommended in students first spring quarter of study at UCSD. This course gives an overview of issues and topics in bioinformatics.

[3] CSE 30 prerequisite will be waived.

[4] New courses for the bioinformatics program: CSE 181 is cross-listed with BIMM 181 and BENG 181; CSE 182 is cross-listed with BIMM 182, Chem 182, and BENG 182; CSE 184 is cross-listed with BIMM 184 and BENG 184; and (BENG 183, BIMM 185, and Math. 186 are also new courses but they are not cross-listed with any other courses).

[5] Students must complete five CSE technical electives from the approved list.

B.A. Computer Science Program

The B.A. computer science program gives students more latitude in designing their course of study. The lower-division program is designed to provide a strong foundation in mathematics, physics, programming methodology and skills, and computer organization. Upper-division core courses deal with the theory and design of algorithms, hardware, and software. Students can gain additional breadth and/or depth in computer science and engineering by an appropriate selection of technical electives. By requiring fewer technical electives, the B.A. computer science program serves those students desiring more time for undergraduate studies outside their major subject.

The department requires a total of 104 units for the B.A. computer science program (not including the general-education requirements). There are three varieties of requirements: lower-division, upper-division, and technical electives.

1. LOWER-DIVISION REQUIREMENTS

Students are expected to complete the following fifty-two units by the end of their sophomore year.

Computer Science and Engineering: CSE 8B or 11, 12, 20 or Math. 15A, CSE 21 or Math. 15B, and CSE 30; twenty units.

Note: Students without any programming experience are advised to take CSE 8A and then CSE 8B, instead of CSE 11. CSE 11 is a faster paced version of CSE 8A and CSE 8B, and requires experience in programming with a compiled language.

Mathematics: Math. 20A-B-C-D and 20F; twenty units.

Physics: Phys. 2A-B-C; twelve units. Math. 20A is a prerequisite for Phys. 2A. Students whose performance on the Department of Mathematics placement test permits them to start with Math. 20B or a higher course may take Phys. 2A in the fall quarter of the freshman year; all others will take Phys. 2A in the winter quarter of the freshman year. Students who received high grades in both calculus and physics in high school may substitute the major's sequence, Phys. 4A-B-C for Phys. 2A-B-C.

2. UPPER-DIVISION REQUIREMENTS

All B.A. computer science students are required to take CSE 100 or Math. 176, CSE 101 or Math. 188, CSE 105 or Math. 166, CSE 120, 131A, 131B, 140, 140L, 141, and 141L; thirty-six units.

Students are expected to complete almost all of these courses by the end of their junior year. If students want to accelerate their program, they should consider taking CSE 100 or Math. 176, CSE 105 or Math. 166, and/or CSE 140 and 140L in the sophomore year.

3. TECHNICAL ELECTIVES

B.A. computer science students are required to take four technical electives for a total of sixteen units. Of these four electives, at least two must be computer science and engineering upper-division or graduate courses.

The remaining two technical electives can be chosen from a wider set of courses that includes computer science and engineering upper-division courses, graduate courses, and other electives as listed under the section titled **Electives**. Other restrictions in the selection of technical electives are also given in the section **Electives**.

4. B.A. COMPUTER SCIENCE, SAMPLE PROGRAM

FALL	WINTER	SPRING
FRESHMAN YEAR		
CSE 8A or CSE 11	CSE 12	CSE 20
Math. 20A	Math. 20B	or Math. 15A
GE	Phys. 2A	Math. 20C
	GE	Phys. 2B
		GE
SOPHOMORE YEAR		
CSE 21 or Math. 15B	Math. 20F	CSE 30
Math. 20D	GE	GE
Phys. 2C	GE	GE
JUNIOR YEAR		
CSE 100 or Math. 176	CSE 101 or Math. 188	CSE 120
CSE 140	CSE 141	CSE Tech. Elec.
CSE 140L	CSE 141L	GE
CSE 105 or Math. 166	GE.	GE
SENIOR YEAR		
CSE 131A	CSE 131B	Tech. Elec.
CSE Tech. Elec.	Tech. Elec.	
GE		

Electives

The discipline of computer science and engineering interacts with a number of other disciplines in a mutually beneficial way. These disciplines include mathematics, electrical engineering, and cognitive science. The following is a list of upper-division courses from these and other disciplines that can be counted as technical electives.

At most four units of CSE 197, 198, or 199 may be used towards technical elective requirements. CSE 195 cannot be used towards course requirements. Undergraduate students must get instructor's permission and departmental stamp to enroll in a graduate course.

Students may not get duplicate credit for equivalent courses. The *UCSD General Catalog* should be consulted for equivalency information and any restrictions placed on the courses. Additional restrictions are noted below. Any deviation from this list must be petitioned.

Mathematics: All upper-division courses except Math. 168A-B, 179A-B (Math. 183—Computer Engineering majors only), 184A-B, 189A-B, and 195–199.

If a student has completed CSE 167, then he or she cannot get elective credit for Math. 155A. Students may receive elective credit for only one of the following courses: CSE 164A, Math. 174, Math. 173, Phys. 105A-B, CENG 100, MAE 107. No credit for any of these courses will be given if Math. 170A-B-C is taken. Students may receive credit for either one of the following: Math. 166 or CSE 105 (but not both), Math. 188 or CSE 101 (but not both), Math. 176 or CSE 100 (but not both).

Credit will be given for only one of the following: ECE 109 or Math. 183 or Econ. 120A.

Electrical and Computer Engineering: All ECE upper-division courses except 195-199.

Students may not get credit for both CSE 123A and ECE 158A or CSE 143 and ECE 165. Credit will be given for only one of the following: ECE 109 or Math. 183 or Econ. 120A.

Cognitive Science: Sensation and Perception 101A; Learning, Memory, and Attention 101B, Language 101C, Distributed Cognition 102A, Cognitive Ethnography 102B, Cognitive Engineering 102C, Neuroanatomy and Physiology 107A, Systems Neuroscience 107B, Cognitive Neuroscience 107C, Programming Methods for Cognitive Science 108D, Neural Network Models of Cognitive I 108E, Advanced Programming Methods for Cognitive Science 108F, Human Computer Interaction 120, Human Computer Interaction Programming 121, Semantics 150, Language Comprehension 153, Natural and Artificial Symbolic Representational Systems 170, Neural Network Models of Cognition II 181, Artificial Intelligence Modeling II 182.

Students may not get credit for both CSE 150 and Advanced Programming Methods for Cognitive Science 108F or for both CSE 151 and Artificial Intelligence Modeling II 182.

Mechanical and Aerospace Engineering: All upper-division MAE courses except MAE 140 (**ONLY** Computer Science majors may take MAE 140) and 195-199.

Students may receive elective credit for only one of the following courses: CSE 164A, Math. 174, Math. 173, Phys. 105A-B, CENG 100, MAE 107. Students may only get credit for one of the two courses, CSE 167 or MAE 152.

Economics: Microeconomics 100A-B, Game Theory 109, Macroeconomics 110A-B, Mathematical Economics 113, Econometrics 120A-B-C, Applied Econometrics 121, Management Science Microeconomics 170A-B, Decisions Under Uncertainty 171, Introduction to Operations Research 172A-B-C, Economic and Business Forecasting 178.

Credit will be given for only one of the following: ECE 109 or Math. 183 or Econ. 120A.

Linguistics: Phonetics 110, Phonology I 111, Phonology II 115, Morphology 120, Syntax I 121, Syntax II 125, Semantics 130, Mathematical Analysis of Language 160, Computers and Language 163, Computational Linguistics 165, Principles of Discourse and Dialog 169, Psycholinguistics 170, Language and the Brain 172, and Sociolinguistics 175.

Engineering: Team Engineering 101.

Music: Computer Music II 172, Audio Production: Mixing and Editing 173.

Psychology: Engineering Psychology 161

Minor and Program of Concentration

The CSE minor requires successful completion of a total of nine CSE courses. To be admitted into the minor, students must have a 2.5 GPA and a C– or better in CSE 8B or 11, 12, 20, 21, 30. The remaining four CSE courses are CSE 100, CSE 101, and two additional CSE upper-division courses subject to enforcement of prerequisites. In order for the minor to be awarded students must receive an average 2.0 GPA in the upper-division courses.

Note: Students without any programming experience are advised to take CSE 8A and then CSE 8B, instead of CSE 11. CSE 11 is a faster paced version of CSE 8A and CSE 8B, and requires experience in programming with a compiled language.

Students should consult their college provost's office concerning the rules for the minor or program of concentration. Because our undergraduate program is highly impacted, winter or spring enrollment is recommended for CSE 8A or CSE 8B or CSE 11.

Effective Fall 2000, Mathematics/Computer Science Majors will not be allowed the Minor in Computer Science.

Computing Courses for Non-Majors

The department offers a slow-pace course providing a practical introduction to computers, computation, and programming: CSE 5A—an introduction to structured programming using the C/Java programming language. We also offer an introduction in "fluency in information technology": CSE 3—an introduction to basic information students need to deal with information technology. It is more of a concepts course than a programming course, but some simple programming will be done as part of the teaching of concepts.

Admission to Major

FRESHMEN STUDENTS

Freshman students who have excelled in high school and have declared CSE on their application will be directly admitted by the dean of the School of Engineering into their major. The only way to become a computer science (CS) or computer engineering (CE) major is to be directly admitted as an entering freshman (transfer students see TRANSFER STUDENTS section below) These students will be notified directly of their status.

Because of heavy student interest in departmental programs, and the limited resources available to accommodate this demand, maintenance of a high quality program makes it necessary to limit enrollments to the most qualified students.

Admission to the department as a major, transfer, or minor is in accordance with the general requirements established by the School of Engineering.

TRANSFER STUDENTS

The B.S. or B.A. in Computer Science and the B.S. in Computer Engineering are heavily impacted majors and admission is limited to applicants who have demonstrated a high level of achievement commensurate with the prospect of success in these majors. Successful applicants must have completed substantial training at the community college and must have achieved a high level of academic performance there. For example, the required minimum of 90 quarter transfer units must include 18 quarter units of calculus, 12 quarter units of calculus-based physics, and the highest level computer science course offered at their community college.

Effective fall 2001 applicants seeking admission as transfer students will be considered for direct admission into the Computer Science (CS) or Computer Engineering (CE) majors in the Department of Computer Science & Engineering (CSE). The only way to become a computer science (CS) or computer engineering (CE) major is to be directly admitted as an entering transfer student. Although the actual required GPA cutoff depends on the number of openings, at least a 3.2 GPA in the community college transfer courses, and a 3.4 GPA in math, physics and computer science courses, are likely to be needed to gain admission. Transfer students who have declared pre-CSE will be considered for direct admission to the major.

Return To Contents

CONTINUING UCSD STUDENTS
Exceptional Admission Program

Space permitting and in its sole discretion, the computer science and engineering department may periodically grant admission to the Computer Science (CS) or Computer Engineering (CE) majors to a small number of academically exceptional UCSD undergraduate students who were not admitted to these majors as entering students. Exceptional admission will be considered for students having an overall UCSD GPA of 3.5 or better who have taken at least two CSE, math, or science courses demonstrating special aptitude for the CS or CE curriculum. Applications for exceptional admission must include submission of a course plan demonstrating ability to satisfy graduation requirements and a personal statement addressing the applicant's motivation to join the CS or CE major, in addition to other criteria established by the department. (For admission into the B.S. in Computer Science with specialization in Bioinformatics, please see the following section: Admission into B.S. in Computer Science with specialization in Bioinformatics).

Admission into B.S. in Computer Science with a specialization in Bioinformatics

Since the number of pre-majors and majors will be limited as described in the section on bioinformatics, student demand may exceed capacity. Therefore, admission to the specialization is not guaranteed and will be based on academic excellence, as described below. Since bioinformatics is an interdisciplinary major, a steering committee involving faculty from the participating departments will select among the best candidates applying and recommended through each department, while insuring active participation of the departments and divisions offering the major. The final decision on admission to the pre-major and major will be made by the Bioinformatics Steering Committee, in consultation with the departments. The application is found at http://www.cse.ucsd.edu/undergrad/admissions/admissionshome.html, or see the CSE department at AP&M 3402.

FRESHMEN AND CONTINUING STUDENTS

Students (freshmen or continuing UCSD students) will be admitted into one of our existing undergraduate majors (B.A. Computer Science, B.S. Computer Science, and B.S. Computer Engineering) through the direct admission process or through the exceptional admission program. Students will then have the option of trying to enter the bioinformatics program by applying for the bioinformatics pre-major (while still retaining their current major status) once they complete the first four screening courses (Math 20B, Math 21C, BILD 1, Chem 6A). Students will then formally apply to the bioinformatics major upon completion of the remaining screening courses CSE 11 and CSE 12. If admitted, students will become bioinformatics majors in CSE. If not, they can continue in their current CSE major.

TRANSFER STUDENTS

Transfer students will be admitted into one of our existing undergraduate majors (B.A. Computer Science, B.S. Computer Science, and B.S. Computer Engineering) through the direct admission process or through the exceptional admission program. Effective fall 2003, CSE transfer students can directly apply to the bioinformatics major if they completed the following courses prior to transferring to UCSD.

- A year of calculus (equivalent to Math 20A-B-C at UCSD)
- A year of general chemistry, with lab (equivalent to Chem 6A, 6B/6BL and Chem 6C at UCSD)
- The highest level programming course offered at the community college (equivalent to CSE 11 and CSE 12 at UCSD)
- One semester of cell biology (equivalent to BILD 1 and BILD 2 at UCSD)

Those who have not completed the above courses may be admitted as bioinformatics pre-majors and will be allowed a maximum of three quarters to complete pre-major requirements. Students will then formally apply to the bioinformatics major upon completion of the remaining screening courses CSE 11 and CSE 12. If admitted, students will become bioinformatics majors in CSE. If not, they can continue in their current CSE major. Transfer students are encouraged to complete these requirements at the community college.

Enrollment in CSE courses

Student demands exceed capacity in many CSE courses. Accordingly, many CSE courses may have enrollment restrictions which give priority to students in the following order:

1. CSE majors, CSE M.S., and CSE Ph.D. students.
2. CSE pre-majors, and ECE CE majors and pre-majors, and Math-CS majors. CSE pre-major status is conferred to transfer students and to those students admitted prior to fall 1998.
3. Students fulfilling a non-elective requirement in another program.
4. CSE minors.
5. All other majors or pre-majors in other SOE departments.
6. All others, with permission of the department of Computer Science and Engineering.

Where these restrictions apply, the registrar will not enroll low-priority students in the course.

ENROLLING IN UPPER-DIVISION COURSES

The Department of Computer Science and Engineering will attempt to provide sufficient sections of all lower-division courses. Students will, however, be screened to ensure that they meet all course enrollment restrictions.

Admission to upper-division courses will be restricted to students having completed all prerequisites with a C- or better (or consent of the instructor.) Courses have enrollment restrictions which give priority in the following order: students admitted by the department to a major or minor curriculum; students fulfilling a requirement for another major; all others. Within these categories, priority is determined on the basis of graduation date and/or credits completed. Where these restrictions apply, the registrar will not enroll nonmajors except by department approval. Students who are undeclared will not be admitted to upper-division computer science and engineering courses.

Those students not in compliance with the above restrictions should be forewarned that they will automatically be dropped from course rosters (at any time during the quarter) when it comes to the attention of the department that a student is enrolled in a course without being eligible because of restrictions and/or the performance standards have not been met. Admission to all computer science and engineering courses will require obtaining either authorization through telephone registration or department stamps on an add/drop card, and it will be given only by the student affairs personnel.

Graduation Requirements

All major requirements and technical electives except CSE 197, 198, or 199 must be taken for a letter grade. To graduate, a grade-point average of 2.0 will be required in upper-division courses in the major, including technical electives. In addition, each student must satisfy general-education course requirements determined by the student's college, as well as major requirements determined by the department. The five colleges at UCSD require widely different numbers of general-education courses. Each student should choose his or her college carefully, considering the special nature of the college and breadth of education, realizing that some colleges require considerably more courses than others.

Five-year Bachelor's—Master's Program

Undergraduate students in the Department of Computer Science and Engineering who are enrolled in the B.S. or B.A. computer science or B.S. computer engineering degree programs, and who have a cumulative GPA of a 3.4 and also a GPA of 3.4 in at least seven core courses, are eligible to apply for the Five Year Bachelor's-Master's Degree Program. The deadline to apply is the fourth week of the fourth quarter before graduation. Acceptance into this program is an honor which carries with it practical benefits—the graduate application process is simplified (no GREs required), students accepted into this program can be admitted fall, winter, and spring quarter, based upon availability of openings in the program. Advanced students are given access to graduate level courses and have the opportunity to do graduate level research earlier under the direct supervision of UCSD's faculty, and students are able to complete the B.A., B.S., and M.S. degree within a five-year time period. Courses taken can be used toward either the B.A., B.S., or M.S. degree, but not counted toward both degrees. Additional information and applications can be obtained by contacting the CSE Student Affairs Office, APM 3402.

The Graduate Program

The graduate program offers master of science and doctor of philosophy degrees in computer science and computer engineering. To be accepted into either course of study, a student must have a B.A./B.S. degree in computer science, computer engineering, or a related area.

The graduate program is concerned with fundamental aspects of computation; emphasis is divided among the areas of theory, hardware, software systems, and artificial intelligence. The computer engineering specialization places a greater emphasis on hardware and the design of computer systems.

Admission to the graduate program is done through the Office of Graduate Admissions, Department of CSE. Deadline for application is January 5. Admissions are always effective the following fall quarter. For admission requirements, please refer to the departmental Web page: http://www-cse.ucsd.edu/graduate/

Admission decisions for the M.S. and Ph.D. programs are made separately. An M.S. student who wishes to enter the Ph.D. program must submit a new application to the CSE admissions committee.

Computer Science Program

Master of Science Program

The department offers the master of science degree in computer science. The degree can be pursued under either the Thesis Plan I or the Comprehensive Examination Plan II. Each plan requires forty-nine units of work. For full-time students, all the requirements must be completed within two years. Students with an adequate background in computer science can complete the M.S. program within four to five quarters of full-time study.

THESIS OR COMPREHENSIVE EXAM

There are two plans of study for the master's degree: Plan I, in which the student writes a thesis, and Plan II, in which the student takes a set of comprehensive exams.

Plan I: Thesis Option, No Comprehensive Exam

This plan involves both course work and research, culminating in the preparation of a thesis. The student must take twelve units of CSE 298 (Independent Study) to fulfill the research requirement, and a thesis based on research must be written and subsequently reviewed by the thesis committee. This committee, which is appointed by the dean of Graduate Studies and Research, consists of three faculty members, with at least two members from within the CSE department.

Plan II: Comprehensive Examination, No Thesis

Under this plan, the student must pass a written comprehensive examination designed to test the student's knowledge in basic computer science material. The examination can normally be passed with a thorough knowledge of topics covered in the undergraduate and first-year graduate computer science programs. It is offered every year in the first few weeks of the fall quarter and in the first few weeks of the spring quarter. Each student is allowed three attempts to pass the examination. The student must secure at least a master's-level pass in the written comprehensive examination. More information regarding the comprehensive examination can be found in a separate document provided by the CSE graduate office.

In particular, the written examination is structured around the three CSE core areas: algorithms and data structures; operating systems; and computer architecture and digital logic design.

Required Courses

Students entering the M.S. Program in Computer Science will choose an area of concentration from among twelve areas. Each concentration is an area in which the faculty has significant research expertise.

The typical concentration is a collection of three courses which are designed to give the student in-depth training in the chosen field. Additionally, to ensure breadth, all students are required to take four core courses.

Core courses must be completed with an average grade of B and no grade below B–. The four core courses required of all students are as follows:

CSE 202. Algorithms Design and Analysis
CSE 221. Operating Systems
CSE 240A. Principles of Computer Architecture
CSE 292. Faculty Research Seminar

The department expects to offer concentrations in the following areas:

Artificial Intelligence
Communication Networks
Computer Architecture and Compilers
Cryptography and Security
Database and Information Retrieval

Design Automation for Microelectronic Designs
Distributed and Fault-Tolerant Computing
Multimedia Systems
Parallel and Scientific Computing
Software Engineering
Storage Systems
Ph.D. Preparation

The specific courses involved in each of the concentrations are detailed in a separate bulletin which is available in the Graduate Student Affairs Office, 3402 AP&M.

Project

Students electing Plan II are required to execute a project while enrolled in four units of CSE 293.

Electives

In addition to completing the required core courses and fulfilling either the thesis or comprehensive examination requirements described above, the student must also complete additional approved courses to bring the total number of units to forty-nine. The number of units of electives depends upon whether the student chooses Plan I or Plan II. The electives consist of other CSE graduate courses or courses from a list of approved electives. Units obtained in the courses CSE 293, 298, 299, 500, 501, 599, and any of the seminar courses CSE 209, 229, 259, 269, 290, and 294 do not count toward the elective requirement.

Computer Engineering Program

Computer engineering, jointly administered between the CSE and ECE departments, offers the master of science and doctoral degrees with the degree title computer science and engineering (computer engineering). Computer engineering explores the engineering analysis and design aspects of algorithms and technology. Specific research areas include computer systems, signal processing systems, architecture, networks, computer-aided design, fault tolerance, and data storage systems.

Master of Science Program

The degree can be pursued under either the Thesis Plan I or the Comprehensive Examination Plan II. Each plan requires forty-nine units of work. For full-time students, all the requirements must be completed within two years. Students with an adequate background in computer engineering can complete the M.S. program within four to five quarters of full-time study.

PLAN I: THESIS OPTION, NO COMPREHENSIVE EXAM

This plan of study involves both course work and research, culminating in the preparation of a thesis. A total of forty-nine units of credit is required, as follows:

Core Courses

The following core courses must be completed with an average grade of B and no grade below B–:

Three Software Courses:
CSE 202
CSE 221
CSE 231

Three Hardware Courses:
CSE 240A
ECE 260A
CSE 243A or ECE 251A or ECE 263A

Two Analysis Courses:
CSE 200
CSE 222A or ECE 257A

and:
CSE 292

Electives

Students must elect at least four technical units among graduate courses within the Depart-ments of AMES, CSE, ECE, Mathematics, and Physics.

Thesis

Twelve units of CSE 298 must be taken with a faculty member in CSE or ECE who agrees to act as adviser for the thesis to fulfill the research requirement.

A thesis based on research must be written and subsequently reviewed by a committee, consisting of three faculty members, with at least two members from within the CSE department. The committee is appointed by the dean of Graduate Studies.

PLAN II: COMPREHENSIVE EXAMINATION, NO THESIS

In order to receive the M.S. degree in computer engineering under this plan, a student must complete the course requirements listed below and pass a written comprehensive examination.

The written examination is structured around the following three CSE core areas: algorithms and data structures; operating systems; and computer architecture and digital logic design.

Core Courses

Three Software Courses:
CSE 202
CSE 221
CSE 231

Three Hardware Courses:
CSE 240A
ECE 260A
CSE 243A or ECE 251A or ECE 263A

Two Analysis Courses:
CSE 200
CSE 222A or CSE 257A

and
CSE 292

Electives

Students must elect at least twelve technical units among graduate courses within the Departments of MAE, CSE, ECE, Mathematics, and Physics.

Project

Four units of CSE 293.

Comprehensive Examination

The comprehensive examination is designed to test the student's knowledge in basic computer science and engineering material. The examination can normally be passed with a thorough knowledge of topics covered in the undergraduate and the first-year graduate computer science or computer engineering programs.

It is offered every year in the first few weeks of the fall quarter and in the first few weeks of the spring quarter. If fewer than seven people sign up, then the department may cancel the examination in the spring quarter. Each student is allowed three attempts to pass the examination. The student must secure at least a master's-level pass in the written comprehensive examination.

This examination is the same for both the computer science and the computer engineering graduate programs. More information about the comprehensive examination can be obtained in a separate document from the CSE graduate office.

Doctoral Programs

Ph.D. Degree Programs in Computer Science and Computer Engineering

CSE offers doctor of philosophy degrees in computer science and in computer engineering, providing a research-oriented education in preparation for a research, industrial, or entrepreneurial career. These programs explore both the fundamental aspects and application of computation, spanning theory, software, hardware, and applications. Our particular areas of research expertise include:

- algorithms
- artificial intelligence
- bioinformatics
- complexity theory
- computer architecture/compilers
- VLSI/CAD and embedded systems
- databases and information management
- distributed systems and networking
- graphics and vision
- high performance computing
- mobile computing
- security and cryptography
- software engineering

COMPETENCY REQUIREMENT

We consider command of the materials covered in the following courses to be an adequate background for the Ph.D. program. The competency requirement is intended to ensure that Ph.D. students already have or acquire this undergraduate background. Students, who do not have this background at the time of entry, may be asked to either enroll in the following undergraduate class or to study it independently and demonstrate their knowledge by obtaining a B+ or better in the class or in the final exam.

CSE 101 (Algorithms)

CSE 105 (Theory of Computation)

CSE 120 (Principles of Computer Operating Systems)

Two of the following three courses:

CSE 130, 131A-B (Programming Languages/Compilers)

CSE 141 (Introduction to Computer Architecture)

COURSE REQUIREMENTS

The course requirement is intended to ensure that students are exposed to fundamental concepts and tools (core requirement), a deep up-to-date view of their research area (depth requirement), and advanced, up-to-date view of the some topics outside their area (breadth requirement). Ph.D. students are expected to complete the core, depth, and breadth requirements in the first two years of the program. All required coursework must be taken for a letter grade except for CSE 291 (Topics in CSE), CSE 292 (Faculty Research Seminar), CSE 299 (Research), and CSE 500 or CSE 501 (Teaching Assistantship) for which only S/U grades are allowed.

Units obtained from a single course cannot count more than once towards satisfying the requirement in each of the core, depth, breadth, and elective areas. Ph.D. students who have taken similar courses elsewhere may petition for a waiver of the required courses or for substitution by alternative courses.

Core Requirements

The core requirements ensure that the Ph.D. students share knowledge of fundamental concepts and tools. Each Ph.D. student must take these courses for letter grade and maintain an overall core course GPA of 3.4 with no grade lower than a B- (except for CSE 292, for which a letter grade is not assigned). A student will typically complete all the core courses within the first year of graduate study, and must complete all core courses within two years of entry.

CSE 202 (Algorithms)

CSE 221 (Operating Systems)

CSE 240A (Architecture)

CSE 200 (Complexity) for Computer Science

or CSE 241A (Introduction to Computing Circuitry) for Computer Engineering

CSE 292 (Faculty Research Seminar)

Depth Requirements

The depth requirement ensures that a Ph.D. student has, early on in his or her career, acquired some depth of knowledge in a general research area. Each Ph.D. student must select one of the following areas as his or her major area. The student must take three courses (12 units) from this major area. The student must take these courses for letter grade and maintain an overall depth course GPA of 3.4 with no grade lower than B- in these courses. One of these three courses can be Topics in CSE (CSE 291) or Independent Study (CSE 299), which are not taken for a letter grade. The department will maintain a list of appropriate courses for each major area.

The major areas are:

Theoretical Computer Science

Programming Languages, Compilers, and Software Engineering

Computer Systems

Database Systems

Computer Engineering

Artificial Intelligence

Breadth Requirements

Research in computer science and engineering is increasingly interdisciplinary, and acquiring a broader view of the field in general is important. Each Ph.D. student must take three courses (12 units) from at least two other areas different from the major area. Courses must be taken for letter grade and students must maintain an overall breadth course GPA of 3.0 with no grade lower than B- in these courses. Units obtained in CSE 209 series, 229 series, 239 series, 249 series, 259 series, 269 series, 279 series, 289 series, 290, 292, 293, 294, 298, 291, 299, 500, 501, and 599 do not count toward the breadth requirement.

Electives

In addition to the above required course work, each student is expected to take two additional courses (8 units). Students must obtain no grade lower than C- in these courses. Undergraduate upper-division courses, CSE 291, and CSE 299 may fulfill this requirement. Units obtained in CSE 209 series, 229 series, 239 series, 249 series, 259 series, 269 series, 279 series, 289 series, 290, 292, 293, 294, 298, 500, 501, and 599 do not count toward the elective requirement.

RESEARCH EXAM REQUIREMENT

The research exam is intended to verify three components of the student's preparation for Ph.D. research: (1) breadth of comprehension sufficient to enable computer science research in areas beyond the topic(s) of the research exam and thesis; (2) ability to perform critical study, analysis, and writing in a focused area; and (3) research experience.

The research exam has both an oral part and a written part. The oral part of the research exam is distinct from, and cannot be combined with the University Qualifying Exam. Grading criteria for each part, and standards for passing, are available from the CSE department graduate office.

The research exam is conducted by a committee of three faculty members approved by the Graduate Committee and the chair of the department. At least two committee members must be CSE senate faculty, and the student's adviser must be a member of the committee. The normative time for passing the research exam is by the end of the second year of study; the exam must be passed by the end of the third year if the student is to continue in the Ph.D. program. Passing the research exam enables a Ph.D. student to receive the M.S. degree. Ph.D. students who do not pass the exam after two attempts will be given the opportunity to write a thesis in order to receive a terminal M.S. degree. The M.S. degree is only granted to those students who do not already hold an M.S. degree prior to entering the CSE department at UCSD.

TEACHING ASSISTANT REQUIREMENT

All students enrolled in the Ph.D. program must have one quarter of training as a teaching assistant. This is a formal degree requirement and must be completed before the student is permitted to graduate. The requirement is met by serving as a 50 percent teaching assistant and taking either CSE 500 or CSE 501(Teaching Assistantship).

QUALIFYING EXAMINATION AND ADVANCEMENT TO CANDIDACY

The qualifying examination is a requirement for advancement to candidacy. Prior to taking the qualifying examination a student must have satisfied the departmental competency, course, and research exam requirements and must have been accepted by a CSE faculty member as a Ph.D. thesis candidate. All doctoral students are expected to advance to candidacy by the end of their third year, and advancement is mandatory by the end of the fourth year. The examination is administered by a doctoral committee appointed by the dean of Graduate Studies and Research and consists of faculty from CSE and other departments. More information on the composition of the committee can be obtained from the CSE graduate office. The examination is taken after the student and his or her adviser have identified a topic for the dissertation and an initial demonstration of feasible progress has been made. The candidate is expected to describe his or her accomplishments to date as well as future work.

DISSERTATION

The dissertation defense is the final Ph.D. examination. A candidate for the Ph.D. is expected to write a dissertation and defend it in an oral examination conducted by the doctoral committee.

DEPARTMENTAL PH.D. TIME LIMIT POLICIES

Students must be advanced to candidacy by the end of four years. Total university support cannot exceed seven years. Total registered time at UCSD cannot exceed eight years.

FINANCIAL AID

Financial support is available to qualified graduate students in the form of fellowships, loans, and assistantships. Anticipated stipends for half-time research assistantships are $1573 per month, with the possibility of full-time employment during the summer months. For a half-time teaching assistantship, the anticipated stipend will be $1571 per month. Requests for application forms for admission and financial support should be directed to the Department of Computer Science and Engineering.

COURSES

Student demand exceeds capacity in many CSE courses. Accordingly, many CSE undergraduate courses may have enrollment restrictions which give priority to students in the following order:

- CSE majors, CSE M.S., and CSE Ph.D. students
- CSE pre-majors, ECE CE majors and pre-majors, and Math-CS majors
- students fulfilling a non-elective requirement in another program
- CSE minors
- all other majors or pre-majors in other SOE departments
- all others, with permission of the Department of Computer Science and Engineering

Where these restrictions apply, the registrar will not enroll low-priority students in the course.

A tentative schedule of course offerings is available from the department, APM 3402, each spring for the following academic year. The tentative schedule for 2002–2003 is also found at http://www-cse.ucsd.edu/ugrad/offerings02-03.html.

LOWER-DIVISION

CSE 3. Fluency in Information Technology (4)
Introduces the concepts and skills necessary to effectively use information technology. Includes basic concepts and some practical skills with computer and networks. *Prerequisite: none.*

CSE 5A. Introduction to Programming I (4)
(Formerly CSE 62A) Introduction to algorithms and top-down problem solving. Introduction to the C language including functions, arrays, and standard libraries. Basic skills for using a PC graphical user interface operating system environment. File maintenance utilities are covered. (A student may not receive credit for CSE 5A after receiving credit for CSE 10 or CSE 11 or CSE 8B or CSE 9B or CSE 62B or CSE 65.) *Prerequisite: A familiarity with high-school level algebra is expected, but this course assumes no prior programming knowledge.*

CSE 8A. Introduction to Computer Science: Java (4)
Basic UNIX. Basics of Java language. Classes, methods, and parameters. Modularity and abstraction. Documentation techniques. Testing and verification techniques. Basic Inheritance. Event driven programming. Programming with AWT library or other similar library. CSE 8A-B is a slower paced version of CSE 11 with more programming practice. (Students who have taken CSE 9A or CSE 9B or CSE 10 or CSE 11 may not take CSE 8A.) *Prerequisite: high school algebra.* Majors only.

CSE 8B. Introduction to Computer Science: Java (4)
Continuation of the Java language. Continuation of programming techniques. More on Inheritance. Exception handling. CSE 8A-B is a slower paced version of CSE 11 with more programming practice. (Students who have taken CSE 9A or CSE 9B or CSE 10 or CSE 11 may not take CSE 8B.) *Prerequisite: high school algebra.* Majors only.

CSE 11. Introduction to Computer Science and Object-Oriented Programming: Java (4)
Introduction to computer science and programming using the Java language. Basic UNIX. Modularity and abstraction. Documentation, testing and verification techniques. Basic object-oriented programming including inheritance and dynamic bind. Exception handling. Event-driven programming. Experience with AWT library or other similar library. *Prerequisites: high-school algebra and a course in programming in a compiled language.* Majors only.

CSE 12. Basic Data Structures and Object-Oriented Design (4)
Basic data structures including stacks, queues, lists, binary trees, hash tables. Basic object-oriented design including encapsulation, polymorphism, classes as the implementation of abstract data types. Memory management, pointers, recursion, and big-o notation. Uses the C/C++ and Java programming language. *Prerequisite: CSE 8B or CSE 11.*

CSE 20. Introduction to Discrete Mathematics (4)
Basic discrete mathematical structure: sets, relations, functions, sequences, equivalence relations, partial orders, and number systems. Methods of reasoning and proofs: propositional logic, predicate logic, induction, recursion, and pigeonhole principle. Infinite sets and diagonalization. Basic counting techniques; permutation and combinations. Applications will be given

to digital logic design, elementary number theory, design of programs, and proofs of program correctness. Credit not offered for both Math. 15A and CSE 20. Equivalent to Math 15A. *Prerequisites: CSE 8A or CSE 8B or CSE 11. CSE 8B or CSE 11 may be taken concurrently with CSE 20/ Math. 15A.*

CSE 21. Mathematics for Algorithms and Systems (4)
This course will provide an introduction to the discrete mathematical tools needed to analyze algorithms and systems. Enumerative combinatorics: basic counting principles, inclusion-exclusion, and generating functions. Matrix notation. Applied discrete probability. Finite automata. Credit not offered for both Math.15B and CSE 21. Equivalent to Math 15B. *Prerequisites: CSE 20 or Math. 15A.*

CSE 30. Computer Organization and Systems Programming (4)
(Formerly CSE 70.) Introduction to organization of modern digital computers—understanding the various components of a computer and their interrelationships. Study of a specific architecture/machine with emphasis on systems programming in C and Assembly languages in a UNIX environment. *Prerequisites: CSE 12 and CSE 20 or Math. 15A; or consent of the instructor.*

CSE 80. UNIX Lab (2)
The objective of the course is to help the programmers create a productive UNIX environment. Topics include customizing the shell, file system, shell programming, process management, and UNIX tools. (P/NP grades only.) *Prerequisite: CSE 8B or CSE 9B or CSE 10 or CSE 11.*

CSE 87. Freshman Seminar (1)
The Freshman Seminar Program is designed to provide new students with the opportunity to explore an intellectual topic with a faculty member in a small seminar setting. Freshman seminars are offered in all campus departments and undergraduate colleges, and topics vary from quarter to quarter. Enrollment is limited to 15 to 20 students, with preference given to entering freshmen. *Prerequisite: none.*

UPPER-DIVISION

CSE 100. Advanced Data Structures (4)
Descriptive and analytical presentation of data structures and algorithms. Lists, tables, priority queues, disjoint subsets, and dictionaries data types. Data structuring techniques include linked lists, arrays, hashing, and trees. Performance evaluation involving worst case, average and expected case, and amortized analysis. Credit not offered for both Math. 176 and CSE 100. Equivalent to Math. 176. *Prerequisites: CSE 12, CSE 21 or Math. 15B, and CSE 30, or consent of the instructor.* Majors only.

CSE 101. Design and Analysis of Algorithms (4)
Design and analysis of efficient algorithms with emphasis of non-numerical algorithms such as sorting, searching, pattern matching, and graph and network algorithms. Measuring complexity of algorithms, time and storage. NP-complete problems. Credit not offered for both Math. 188 and CSE 101. Equivalent to Math 188. *Prerequisites: CSE 12, CSE 21 or Math. 15B or Math. 100A or Math. 103A.* Majors only.

CSE 102. Storage System Architectures (4)
Descriptive and analytic introduction to disk drive storage media, external data structures, and their algorithms. Disk drives, external sorting, index structures, disk arrays, reliability, data declustering, and video servers. Merge sort, B+trees, linear hashing, improved RAID data organizations, and SID data layout. *Prerequisite: CSE 100 or Math. 176, CSE 120, CSE 123A, or consent of the instructor.* Majors only.

CSE 105. Theory of Computability (4)
An introduction to the mathematical theory of computability. Formal languages. Finite automata and regular expression. Push-down automata and context-free languages. Computable or recursive functions: Turing machines, the halting problem. Undecidability. Credit not offered for both Math. 166 and CSE 105. Equivalent to Math 166. *Prerequisites: CSE 12, CSE 21 or Math. 15B or Math. 100A or Math. 103A.* Majors only.

CSE 107. Introduction to Modern Cryptography (4)
Topics include private and public-key cryptography, block ciphers, data encryption, authentication, key distribution and certification, pseudorandom number generators, design and analysis of protocols, zero-knowledge proofs, and advanced protocols. Emphasizes rigorous mathematical approach including formal definitions of security goals and proofs of protocol security. *Prerequisites: CSE 21 or Math. 15B, CSE 101 or Math. 188, CSE 105 or Math. 166.* Majors only.

CSE 111. Object Oriented Software Design (4)
Introduction to object-oriented analysis and design. Object-oriented modeling methods for analysis and design, object-oriented general design paradigms, object-oriented design techniques. Cyclic development of object-oriented systems. *Prerequisites: CSE 8B or CSE 9B or CSE 10 or CSE 11, CSE 12, and CSE 100 or Math. 176.* Majors only.

CSE 112. Software Engineering (4)
(Formerly CSE 110.) This course will cover software engineering topics associated with large systems development such as requirements and specifications, testing and maintenance, and design. *Prerequisites: CSE 111.* Majors only.

CSE 120. Principles of Computer Operating Systems (4)
(Formerly CSE 171A.) This course introduces the basic concepts used to structure computer operating systems. Examples of notions introduced and discussed are batch processing, multiprogramming, input/output, pooling, interrupt handling, processes, descriptors, process synchronization, interprocess communication, memory management, virtual memory, caching, buffers, naming, files, interactive command interpreters, and processor scheduling. *Prerequisites: CSE 100 or Math. 176 and CSE 141.* Majors only.

CSE 121. Operating Systems: Architecture and Implementation (4)
(Formerly CSE 171B.) Case study of architecture and implementation of a selected modern operating system. In-depth analysis through a detailed study of source code. Topics include process creation, context-switching, memory allocation, synchronization mechanisms, interprocess communication, I/O buffering, device drivers, and file systems. *Prerequisite: CSE 120.*

CSE 123A. Computer Networks (4)
Introduction to concepts, principles, and practice of computer communication networks with examples from existing architectures, protocols, and standards with special emphasis on the Internet protocols. Layering and the OSI model; physical and data link layers; local and wide area networks; datagrams and virtual circuits; routing and congestion control; internetworking. Transport protocols. *Prerequisite: CSE 120 or consent of the instructor.* Majors only.

CSE 123B. Communications Software (4)
Protocol software structuring, The Internet protocol suite, Inter-process communication, Protocols for real-time and multimedia (digital audio and video) communication, multicast, bridging, and group communication protocols, protocols for mobile and personal communication networks, application-level protocols, secure communication. *Prerequisite: CSE 120 or consent of the instructor. CSE 123A is strongly recommended.* Majors only.

CSE 126. Multimedia Systems (4)
Multimedia technologies; multimedia storage models and structures; data models and interfaces; multimedia information systems; video/audio networking; media synchronization; image computing and information assimilation; conferencing paradigms and structured interaction support. *Prerequisite: CSE 120 or consent of the instructor.*

CSE 127. Introduction to Computer Security (4)
Topics include basic cryptography, security/threat analysis, access control, auditing, security models, distributed systems security, and theory behind common attack and defense techniques. The class will go over formal models as well as the bits and bytes of security exploits. *Prerequisite: CSE 21 or Math. 15B, and CSE 120.* Majors only.

CSE 128. Concurrency (4)
Specification of concurrent programs safety, liveness, and fairness: producer-consumer; mutual exclusion; atomic read/writes; semaphores; monitors; distributed algorithms and memory coherency; programming with threads; concurrency in popular programming languages and operating systems. *Prerequisite: CSE 120.* Majors only.

CSE 130. Programming Languages: Principles and Paradigms (4)
(Formerly CSE 173.) Introduction to programming languages and paradigms, the components that comprise them, and the principles of language design, all through the analysis and comparison of a variety of languages (e.g., Pascal, Ada, C++, PROLOG, ML.) Will involve programming in most languages studied. *Prerequisites: CSE 12 and CSE 100 or Math. 176.* Majors only.

CSE 131A. Compiler Construction I (4)
(Formerly CSE 163A.) Introduction to the compilation of programming languages, principles and practice of lexical and syntactic analysis, error analysis, syntax-directed translation, and type checking. *Prerequisites: CSE 30, CSE 100 or Math. 176, and CSE 105 or Math. 166.* Majors only.

CSE 131B. Compiler Construction II (4)
(Formerly CSE 163B.) Principles and practice for the design and implementation for the back-end of translators for programming languages, symbol tables, syntax-directed translation, code generation, optimization, and compiler structure. *Prerequisites: CSE 30, CSE 100 or Math. 176, CSE 105 or Math. 166, and CSE 131A.* Majors only.

CSE 132A. Database System Principles (4)
Basic concepts of databases, including data modeling, relational databases, query languages, optimization, dependencies, schema design, and concurrency control. Exposure to one or several commercial database systems. Advanced topics such as deductive and object-oriented databases, time allowing. *Prerequisite: CSE 100 or Math. 176.* Majors only.

CSE 132B. Database Systems Applications (4)
Design of databases, transactions, use of trigger facilities and datablades. Performance measuring, organization of index structures. *Prerequisite: CSE 132 or CSE 132A or equivalent.*

CSE 133. Information Retrieval (4)
(Formerly CSE 181.) How to find "relevant" documents (e.g., an electronic mail message or a book) from very large corpora (e.g., all the world's electronic mail or the library.) Students will construct and experimentally evaluate a complete IR system for a modest textual corpus. *Prerequisite: CSE 100 or Math. 176.* Majors only.

CSE 134A. Web Server Languages (4)
Design and implementation of interactive World Wide Web documentation using server-side programs. Languages covered include HTML, Perl, and JavaScript. Other languages as time allows. *Prerequisite: CSE 100 or Math. 176.* Majors only.

CSE 134B. Web Client Languages (4)
Design and implementation of interactive World Wide Web clients using helper applications and plug-ins. The main language covered will be Java. *Prerequisite: CSE 100 or Math. 176.* Majors only.

CSE 140. Components and Design Techniques for Digital Systems (4)
(Formerly CSE 170A) Design of Boolean logic and finite state machines; two-level, multi-level combinational logic design, combinational modules and modular networks, Mealy and Moore machines, analysis and synthesis of canonical forms, sequential modules. *Prerequisites: CSE 20 or Math. 15A, CSE 30. CSE 140L must be taken concurrently.* Majors only.

CSE 140L. Digital Systems Laboratory (2)
(Formerly CSE 175B) Implementation with computer-aided design tools for combinational logic minimization and state machine synthesis. Hardware construction of a small digital system. *Prerequisites: CSE 20, CSE 30. CSE 140 must be taken concurrently.*

CSE 141. Introduction to Computer Architecture (4)
Introduction to computer architecture. Computer system design. Processor design. Control design. Memory systems. *Prerequisites: CSE 140, CSE 140L, or consent of the instructor. CSE 141L should be taken concurrently.* Majors only.

CSE 141L. Project in Computer Architecture (2)
Hands-on computer architecture project aiming to familiarize students with instruction set architecture, and design of process. Control and memory systems. *Prerequisites: CSE 140, CSE 140L, or consent of the instructor. CSE 141 should be taken concurrently.* Majors only.

CSE 142. Advanced Digital Logic Design (4)
(Formerly CSE 170C) Digital logic optimization; functional decomposition and symmetric functions; reliable design and fault diagnosis; structure of sequential machines; asynchronous circuit design. Assignments using logic synthesis tools. *Prerequisites: CSE 140, CSE 140L*

CSE 144. Computer-Aided Design of VLSI Circuits (4)
(Formerly CSE 172B) Introduction to Computer-Aided Design. Placement, assignment and floor planning techniques. Routing. Symbolic layout and compaction. Module generation and silicon compilation. *Prerequisites: CSE 140 and CSE 140L, or consent of the instructor.*

CSE 150. Programming Languages for Artificial Intelligence (4)
Note: CSE 150 is pending CEP approval. (Formerly CSE 162) Experience using two very different approaches to artificial intelligence programming. Symbolic manipulation using LISP, with examples drawn from heuristic search, inference, and/or resolution theorem proving. Pattern recognition and transformation using neural networks with perception and back propagation learning algorithms, applied to problems such as face recognition, English past tense formation, etc. *Prerequisites: CSE 8B or CSE 9B or CSE 10 or CSE 11, CSE 12, and CSE 100 or Math. 176.* Majors only.

CSE 151. Introduction to Artificial Intelligence (4)
An introduction to theoretical issues and computational techniques arising from a comparison of human and machine intelligences. Knowledge representation languages; problem-solving heuristics; machine learning and application areas including vision, robotics, and natural language understanding will be reviewed. *Prerequisite: CSE 150 or consent of the instructor.* Majors only.

CSE 160. Introduction to High Performance Parallel Computation (4)
Introduction to HPPC: parallel architecture, algorithms, software and problem-solving techniques. Areas covered: Flynn's taxonomy, processor-memory organizations, shared and non-shared memory models; message passing and multithreading, data parallelism; speedup, efficiency, and Amdahl's law, communication and synchronization, isoefficiency, and scalability. Topics: run time software techniques, compilers, and grid computing. Assignments given to provide practical experience. *Prerequisite: CSE 100 or Math. 176.* Majors only.

CSE 166. Image Processing (4)
Principles of image formation, analysis, and representation. Image enhancement, restoration, and segmentation; stochastic image models. Filter design, sampling, Fourier and wavelet transforms. Selected applications in computer graphics and machine vision. *Prerequisites: Math. 20F, CSE 100 or Math. 176.* Majors only.

CSE 167. Computer Graphics (4)
(Formerly CSE 177) Representation and manipulation of pictorial data. Two-dimensional and three-dimensional transformations, curves, surfaces. Projection, illumination, and shading models. Raster and vector graphic I/O devices; retained-mode and immediate-mode graphics software systems and applications. *Prerequisites: Math. 2EA/20F and CSE 100 or Math. 176.* Majors only.

CSE 171. User Interface Design (4)
Explores usability, representation and coordination issues in user interface design with some focus on distributed cooperative work, semiotics, and the interplay between socio-cognitive and technical issues. Most examples and homework involve the Web. *Prerequisites: CSE 8B or 11, CSE 20 or Math. 15A, and CSE 100 or Math. 176.* Majors only.

CSE 175. Social and Ethical Issues in Information Technology (4)
Social aspects of information technology, with an emphasis on ethical issues. Topics include ethical theories, privacy and security, spam, e-commerce, the digital divide, open source software, medical informatics, actor-network theory, and some neo-classical economics. *Prerequisites: CSE 100 or Math. 176.* Majors only.

CSE 181. Molecular Sequence Analysis (4)
This course covers the analysis of nucleic acid and protein sequences, with an emphasis on the application of algorithms to biological problems. Topics include sequence alignments, database searching, comparative genomics, and phylogenetic and clustering analyses. Pairwise alignment, multiple alignment, DNS sequencing, scoring functions, fast database search, comparative genomics, clustering, phylogenetic trees, gene finding/DNA statistics. *Prerequisites: CSE 100 or Math. 176, CSE 101 or Math. 188, BIMM 100 or Chem. 114D.* Bioinformatics majors only. CSE 181 is crosslisted with BIMM 181 and BENG 181.

CSE 182. Biological Databases (4)
This course provides an introduction to the features of biological data, how that data are organized efficiently in databases, and how existing data resources can be utilized to solve a variety of biological problems. Relational databases, object oriented databases, ontologies, data modeling and description, survey of current biological database with respect to above, implementation of a database focused on a biological topic. *Prerequisites: CSE 100 or Math. 176.* Bioinformatics majors only. CSE 182 is crosslisted with BIMM 182, Chem 182, and BENG 182.

CSE 184. Computational Molecular Biology (4)
This advanced course covers the application of machine learning and modeling techniques to biological systems. Topics include gene structure, recognition of DNA and protein sequence patterns, classification, and protein structure prediction. Pattern discovery, hidden markov models/support vector machines/neural network/profiles, protein structure prediction, functional characterization of proteins, functional genomics/proteomics, metabolic pathways/gene networks. *Prerequisites: BIMM 181 or BENG 181 or CSE 181, BIMM 182 or BENG 182 or CSE 182.* Bioinformatics majors only. CSE 184 is crosslisted with BIMM 184 and BENG 184.

CSE 190. Topics in CSE (4)
Topics of special interest in computer science and engineering. Topics may vary from quarter to quarter. May be repeated for credit with the consent of instructor. *Prerequisite: department stamp required.*

CSE 191. Seminar in CSE (1–4)
A seminar course on topics of current interest. Students, as well as, the instructor will be actively involved in running the course/class. This course cannot be counted toward a technical elective. *Prerequisite: consent of instructor.*

CSE 195. Teaching (4)
Teaching and tutorial assistance in a CSE course under the supervision of the instructor. (P/NP grades only.) *Prerequisite: consent of the department chair. Department stamp required.*

CSE 197. Field Study in Computer Science and Engineering (4, 8, 12, or 16)
Directed study and research at laboratories away from the campus. (P/NP grades only.) *Prerequisite: consent of the instructor and approval of the department. Departament stamp required.*

CSE 198. Directed Group Study (2 or 4)
Computer science and engineering topics whose study involves reading and discussion by a small group of students under the supervision of a faculty member. (P/NP grades only.) *Prerequisite: consent of the instructor. Department stamp required.*

CSE 199. Independent Study for Undergraduates (2 or 4)
Independent reading or research by special arrangement with a faculty member. (P/NP grades only.) *Prerequisite: consent of the instructor. Department stamp required.*

GRADUATE

CSE 200. Computability and Complexity (4)
Computability review, including halting problem, decidable sets, r.e. sets, many-one reductions; TIME(t(n)), SPACE(s(n)) and general relations between these classes; L, P, PSPACE, NP; NP—completeness; hierarchy theorems; RP, BPP. *Prerequisite: CSE 105 or equivalent.*

CSE 201A. Advanced Complexity (4)
Polynomial-time hierarchy (PH); BPP in second level of PH; Savitch's theorem; NL=coNL; non-uniform and circuit complexity; some circuit lower bounds; IP=PSPACE; probabilistic proof checking (PCP); Application of PCP to approximation hardness; Complexity of proof systems; Parallel complexity classes NC and AC; P-completeness. *Prerequisite: CSE 200.*

CSE 202. Algorithm Design and Analysis (4)
The basic techniques for the design and analysis of algorithms. Divide-and-conquer, dynamic programming, data structures, graph search, algebraic problems, randomized algorithms, lower bounds, probabilistic analysis, parallel algorithms. *Prerequisite: CSE 101 or equivalent.*

CSE 203A. Advanced Algorithms (4)
Modern advances in design and analysis of algorithms. Exact syllabus varies. Topics include approximation, randomized algorithms, probabilistic analysis, heuristics, online algorithms, competitive analysis, models of memory hierarchy, parallel algorithms, number-theoretic algorithms, cryptanalysis, computational geometry, computational biology, network algorithms, VLSI CAD algorithms. *Prerequisite: CSE 202.*

CSE 204A. Combinatorial Optimization (4)
Linear programming, simplex method, duality, and column generating technique. Integer programming introduced via the Knapsack problem. The periodic nature of all integer programs. Why the round-off technique cannot work in general. The solution of the Knapsack problem in polonominal time since nonbasic columns form a group. *Prerequisites: CSE 202 or Linear Algebra or consent of instructor.*

CSE 205A. Logic in Computer Science (4)
(Formerly CSE 208D) Mathematical logic as a tool in computer science. Propositional logic, resolution, first-order logic, completeness and incompleteness theorems with computational viewpoint, finite model theory, descriptive complexity, logic programming, non-monotonic reasoning, temporal logic. Applications to databases, automatic theorem proving, program verification, and distributed systems. *Prerequisite: CSE 200 or consent of instructor.*

CSE 206A. Lattice Algorithms and Applications (4)
(Formerly CSE 207C) Introduction to the algorithmic theory of point lattices (A.K.A. algorithmic geometry of numbers), and some of its most important applications in cryptography and cryptanalysis. Topics usually include: LLL basis reduction algorithm, cryptanalysis of broadcast RSA, hardness of approximating lattice problems. *Prerequisites: CSE 202, CSE 200, or concurrent.*

CSE 206B. Algorithms in Computational Biology (4)
(Formerly CSE 257B) The course focuses on algorithmic aspects of modern bioinformatics and covers the following topics: computational gene hunting, sequencing, DNA arrays, sequence comparison, pattern discovery in DNA, genome rearrangements, molecular evolution, computational proteomics, and others. *Prerequisite: CSE 202 or consent of instructor.*

CSE 207. Modern Cryptography (4)
Private and public key cryptography, introduction to reduction based proofs of security, concrete security, block ciphers, pseudorandom functions and generators, symmetric encryption, asymmetric encryption, computational number theory, RSA and discrete log systems, message authentication, digital signatures, key distribution and key management. *Prerequisites: CSE 202 or consent of instgructor.*

CSE 207C. Lattices and Cryptology (4)
Renumbered to CSE 206A (see above).

CSE 208. Advanced Cryptography (4)
Zero-knowledge, secure computation, session-key distribution, protocols, electronic payment, one-way functions, trapdoor permutations, pseudorandom bit generators, hardcore bits. *Prerequisites: CSE 202, CSE 200, and CSE 207 or consent of instructor.*

CSE 208D. Logic in Computer Science (4)
Renumbered to CSE 205A (see above).

CSE 209A. Topics/Seminar in Algorithms, Complexity, and Logic (1-4)
Topics of special interest in algorithms, complexity, and logic to be presented by faculty and students under faculty direction. Topics vary from quarter to quarter. May be repeated for credit. *Prerequisites: consent of instructor.*

CSE 209B. Topics/Seminar in Cryptography (1-4)
Topics of special interest in cryptography to be presented by faculty and students under faculty direction. Topics vary from quarter to quarter. May be repeated for credit. *Prerequisite: consent of instructor.*

CSE 210. Principles of Software Engineering (4)
(Formerly CSE 264A.) General principles in modern software engineering. Both theoretical and practical topics are covered. Theoretical topics include proofs of correctness, programming language semantics, and theory of testing. Practical topics include structured programming, modularization techniques, design of languages for reliable programming, and software tools. *Prerequisites: CSE 100, 131A, 120, or consent of instructor.*

CSE 211. Software Testing and Analysis (4)
Survey of testing and analysis methods. Introduction to advanced topics in area as well as traditional production methods. Topics include inspections and reviews, formal analysis, verification and validation standards, non-statistical testing, statistical-testing and reliability models, coverage methods, testing and analysis tools, and organization management and planning. Methods special to special development approaches such as object-oriented testing will also be described. *Prerequisite: undergraduate major in computer science or extensive industrial experience.*

CSE 218. Advanced Topics in Software Engineering (4)
This course will cover a current topic in software engineering in depth. Topics in the past have included software tools, impacts of programming language design, and software system structure. (S/U grades permitted.) *Prerequisite: none.*

CSE 221. Operating Systems (4)
Operating system structures, concurrent computation models, scheduling, synchronization mechanisms, address spaces, memory management protection and security, buffering, streams, data-copying reduction techniques, file systems, naming, caching, disk organization, mapped files, remote file systems, case studies of major operating systems. *Prerequisites: CSE 120 and 121, or consent of instructor.*

CSE 222. Communication Networks (4)
Renumbered to CSE 222A (see below).

CSE 222A. Computer Communication Networks (4)
(Formerly CSE 222.) Computer communication network concepts, protocols, and architectures, with an emphasis on an analysis of algorithms, protocols, and design methodologies. Topics will include layering, error control, flow control, congestion control, switching and routing, quality of service management, mobility, naming, security, and selected contemporary topics. *Prerequisite: CSE 123A or consent of instructor.*

CSE 222B. Internet Algorithmics (4)
(Formerly CSE 228H.) Techniques for speeding up Internet implementations including system restructuring, new algorithms, and hardware innovations. Topics include: models for protocols, systems and hardware; efficiency principles; applying these principles to deriving techniques for efficient implementation of common endnode and router functions. *Prerequisites: CSE 123A or CSE 222A or consent of instructor.*

CSE 223. Distributed Systems (4)
Basic structuring concepts: service, server client-server relations, basic network architecture and point-to-point communication services, variable communication delays and failures, logical and physical time, time services, request/reply transport services, remote procedure calls, naming and directory services, distributed concurrency control, distributed file and database services, transactions and the atomic commit problem, security in distributed systems. *Prerequisite: CSE 221 or consent of instructor.*

CSE 224. Computer System Performance Evaluation (4)
Design, measurement, simulation, and modeling for system performance evaluation. Measurement tools such as workloads, benchmarks, experimental design: confidence intervals, analysis of data; simulation: trace driven, Monte Carlo, transient removal; modeling: Little's Law, queuing, mean-value analysis. *Prerequisite: consent of instructor.*

CSE 225. High Performance Distributed Computing (and Computational Grids) (4)
Architecture of high performance distributed systems (e.g., frameworks and middleware). High performance distributed objects (DCOM, Corba, Java Beans) and networking with crosscut issues for performance, availability, and performance predictability. Scalable servers, metacomputing, and scientific computing. *Prerequisites: CSE 121 and CSE 123A or consent of instructor.*

CSE 226. Storage Systems (4)
(Formerly CSE 228B.) Secondary and tertiary storage systems, optical and magnetic media, performance analysis, modeling, reliability, redundant arrays of inexpensive disks, striping, log and maximum distance separable data organizations, sparing. *Prerequisite: CSE 221 or consent of instructor.*

CSE 227. Computer Security (4)
Security and threat models, risk analysis, authentication and authorization, auditing, operating systems security, access control mechanisms, protection mechanisms, distributed systems/network security, security architecture, electronic commerce security mechanisms, security evaluation. *Prerequisite: CSE 221 or consent of instructor.*

CSE 228. Multimedia Systems (4)
(Formerly 228F) Emerging multimedia technologies; multimedia storage models and structures; video/audio networking; intra-media continuity; inter-media synchronization; admission control and support for real time; distributed multimedia systems; structured interaction support (collaboration and teamwork); multimedia encoding. *Prerequisite: consent of instructor.*

CSE 228B. Storage Systems (4)
Renumbered to CSE 226 (see above).

CSE 228F. Multimedia Systems (4)
Renumbered to CSE 228 (see above).

CSE 228H. Internet Algorithmics (4)
Renumbered to CSE 222B (see above).

CSE 229A. Topics/Seminar in Computer Systems (1-4)
Discussion on problems of current research interest in computer systems. Possible areas of focus include: distributed computing, computational grid, operating systems, fault-tolerant computing, storage systems, system services for the World Wide Web. Topics to be presented by faculty and students under faculty direction. Topics vary from quarter to quarter. May be repeated for credit. *Prerequisite: consent of instructor.*

CSE 229B. Topics/Seminar in Networks and Communication (1-4)
Discussion on problems of current research interest in computer networks and communication. Possible areas of focus include: wide-area networking, wireless networks, the Internet, computational grid, operating systems, fault-tolerant computing, storage systems. Topics to be presented by faculty and students under faculty direction. Topics vary from quarter to quarter. May be repeated for credit. *Prerequisite: consent of instructor.*

CSE 229C. Topics/Seminar in Computer Security (1-4)
Discussion on problems of current research interest in computer security. Topics to be presented by faculty and students under faculty direction. Topics vary from quarter to quarter. May be repeated for credit. *Prerequisite: consent of instructor.*

CSE 230. Principles of Programming Languages (4)
(Formerly CSE 273.) Functional versus imperative programming. Type systems and polymorphism; the ML language. Higher order functions, lazy evaluation. Abstract versus concrete syntax, structural and well-founded induction. The lambda calculus, reduction strategies, combinators. Denotational semantics, elementary domain theory. *Prerequisite: CSE 130 or equivalent, or consent of instructor.*

CSE 231. Advanced Compiler Design (4)
(Formerly CSE 264C.) Advanced material in programming languages and translator systems. Topics include compilers, code optimization, and debugging interpreters. *Prerequisites: CSE 100, 131A-B, or consent of instructor.*

CSE 232. Principles of Database Systems (4)
(Formerly CSE 264D.) Database models including relational, hierarchic, and network approaches. Implementation of databases including query languages and system architectures. *Prerequisite: CSE 100 or consent of instructor.*

CSE 232B. Database System Implementation (4)
A hands-on approach to the principles of databases implementation. Algebraic rewriters/optimizers, query processors, triggers. Beyond centralized relational databases. *Prerequisites: CSE 232.*

CSE 233. Database Theory (4)
Theory of databases. Theory of query languages, dependency theory, deductive databases, incomplete information, complex objects, object-oriented databases, and more. Connections to logic and complexity theory including finite model theory and descriptive complexity. *Prerequisite: CSE 200.*

CSE 240. Principles in Computer Architecture I (4)
Renumbered to CSE 240A (see below).

CSE 240A. Principles of Computer Architecture (4)
(Formerly CSE 240.) This course will cover fundamental concepts in computer architecture. Topics include instruction set architecture, pipelining, pipeline hazards, bypassing, dynamic scheduling, branch prediction, superscalar issue, memory-hierarchy design, advanced cache architectures, and multiprocessor architecture issues. *Prerequisite: CSE 141 or consent of instructor.*

CSE 241. Advanced Computer Architecture (4)
Renumbered to CSE 240B (see below).

CSE 240B. Advanced Computer Architecture (4)
(Formerly CSE 241.) This course covers advanced topics in computer architecture, including multiprocessor architecture, interconnection networks, cache coherence, cache consistency. It incorporates the latest research and development on topics such as branch prediction, instruction-level parallelism, multithreading, and cache hierarchy design. *Prerequisite: CSE 240A or consent of instructor.*

CSE 241A. Introduction to Computing Circuitry (4)
Integrated-circuit building blocks of computing systems, and impact on system implementation choices. Devices and interconnects, clocking, basic circuit types, power/ground distribution, arithmetic modules, memories. Design methods covering combinational logic networks, sequential machines, and basic subsystems. Tradeoffs in system implementation. *Prerequisite: CSE 140 or consent of instructor.*

CSE 242A. Integrated Circuit Layout Automation (4)
Couplings among timing, circuits and spatial embedding in nanometer-scale CMOS design. The role, and key problems, of physical layout in IC implementation. Example topics: RTL-to-GDSII methodologies, analyses and estimations, partitioning, floor planning, placement, routing, special net routing, cell generation, compaction. *Prerequisite: CSE 241A or consent of instructor.*

CSE 243A. Introduction to Synthesis Methodologies in VLSI CAD (4)
Hardware software co-design, architectural level synthesis, control synthesis and optimization, scheduling, binding, register and bus sharing, interconnect design, module selection, combinational logic optimization, state minimization, state encoding, and retiming. *Prerequisite: CSE 241A or consent of instructor.*

CSE 244A. VLSI Test (4)
Design for test, testing economics, defects, failures and faults, fault models, fault simulation, automatic test pattern generation, functional testing, memory, PLA, FPGA, microprocessor test, and fault diagnosis. *Prerequisite: CSE 241A or consent of instructor.*

CSE 244B. Testable and Fault Tolerant Hardware Design (4)
Scan path design, BIST architectures, test point insertion, self-checking circuits, test and fault tolerance in architectural synthesis, reconfigurable fault tolerant hardware, and SOC test design. *Prerequisite: CSE 244A or consent of instructor.*

CSE 245. Computer Aided Circuit Simulation and Verification (4)
This course is about the computer algorithms, techniques, and theory used in the simulation and verification of electrical circuits. *Prerequisite: CSE 241A or consent of instructor.*

CSE 246. Computer Arithmetic Algorithms and Hardware Design (4)
Number representation, fixed point adders, subtracters, and multipliers; modified booth's recoding, high-radix multiplication, (non)restoring dividers, SRT division, high-radix dividers, division by convergence, square-rooting, floating point arithmetic, rounding schemes, errors and error control, and floating point adders, subtracters, multipliers, dividers. *Prerequisite: CSE 241A or consent of instructor.*

CSE 247. Application Specific and Reconfigurable Computer Architecture (4)
This course covers architecture concepts used to tailor processors to a specific application or sets of applications. It covers Field-Programmable Gate Arrays (FPGAs), various forms of Application Specific Integrated Circuit (ASIC) designs, Application Specific Integrated Processors (ASIP), and augmenting customizable VHDL cores. *Prerequisite: CSE 241A or consent of instructor*.

CSE 248. Algorithmic and Optimization Foundations for VLSI CAD (4)
Algorithmic techniques and optimization frameworks for large-scale, difficult optimizations. Primal-dual multicommodity flow approximations, approximations for geometric and graph Steiner formulations, continuous placement optimization, heuristics for Boolean satisfiability, multilevel methods, semidefinite programming, and application to other formulations (e.g., scheduling). *Prerequisites: CSE 241A or CSE 242A or consent of instructor.*

CSE 249A. Topics/Seminar in Computer Architecture (1-4)
Topics of special interest in computer architecture to be presented by faculty and students under faculty direction. Topics vary from quarter to quarter. May be repeated for credit. *Prerequisite: consent of instructor.*

CSE 249B. Topics/Seminar in VLSI (1-4)
Topics of special interest in VLSI to be presented by faculty and students under faculty direction. Topics vary from quarter to quarter. May be repeated for credit. *Prerequisite: consent of instructor.*

CSE 249C. Topics/Seminar in CAD (1-4)
Topics of special interest in CAD to be presented by faculty and students under faculty direction. Topics vary from quarter to quarter. May be repeated for credit. *Prerequisite: consent of instructor.*

CSE 250A. Artificial Intelligence I (4)
(Formerly CSE 278A.) Issues in knowledge representation (using logic, semantic networks, production systems, and connectionist representations) will be the focus of this course. A discussion of logic programming languages (like PROLOG) and automatic theorem proving will then lead to a discussion of heuristic search. *Prerequisite: CSE 151 or equivalent.*

CSE 250B. Artificial Intelligence II (4)
(Formerly CSE 278B.) This course will discuss knowledge representations used to search for solutions, make deductions, plan, and problem solve. The application of these techniques to expert systems will be mentioned. Machine learning will also be a major topic of this course. *Prerequisite: CSE 250A.*

CSE 252. Computer Vision (4)
(Formerly CSE 281M.) Illuminant, surface, and camera models. The role of irradiance, chrominance, stereo disparity, optical flow, and texture in computing interpretations of images. Edge detection, image segmentation, local and global constraints from segment boundaries. Object representations and algorithms for recognition. Extremum problems in vision, including regularization and maximum-likelihood techniques. Relation to human vision. *Prerequisite: Math. 2A-B-C-D-E or equivalent.* (S/U grades permitted.)

CSE 252C. Selected Topics in Vision and Learning (4)
Selected topics in computer vision and statistical pattern recognition, with an emphasis on recent developments. Possible topics include: grouping and segmentation, object recognition and tracking, multiple view geometry, kernel-based methods, dimensionality reduction, and mixture models. *Prerequisite: CSE 252 or equivalent and CSE 250B or equivalent.*

CSE 253. Neural Networks (4)
This course covers Hopfield networks, application to optimization problems, layered perceptrons, recurrent networks, and unsupervised learning. Programming exercises explore model behavior, with a final project on a cognitive science, artificial intelligence, or optimization problem of the student's choice. *Prerequisites: knowledge of C and consent of instructor.* (S/U grades permitted.)

CSE 254. Machine Learning (4)
(Formerly CSE 281T.) This course will discuss a wide range of techniques used to allow computers to learn directly from experience with their environment rather than requiring programming by humans. The survey will span both high- and low-level learning techniques as well as theoretical models that allow these various techniques to be compared. (S/U grades permitted.) *Prerequisite: 250B.*

CSE 255. Intelligent Systems (4)
Basic knowledge representation and problem-solving method. Expert system architectures, languages, and tools. Scheduling, planning, diagnosis, and training applications. Fuzzy logic and heuristic control. Neural network, decision tree, and statistical methods for data mining. Guidelines for successful intelligent system deployment. (S/U grades permited.) *Prerequisite: CSE 151 or graduate standing in the Advanced Manufacturing Program, or consent of instructor.*

CSE 256. Statistical Natural Language Processing (4)
An introduction to modern statistical approaches to natural language processing: part of speech tagging, work sense disambiguation and parsing, using Markov models, hidden Markov models and probabilistic context free grammars. *Prerequisite: CSE 250B or equivalent experience.*

CSE 257. Computational Biology (4)
Computational methods are indispensable to an understanding of the vast datasets emerging from human and other organisms' genomes. This course surveys algorithms underlying genome analysis, sequence alignment, phylogenetic trees, protein folding, gene expression, metabolic pathways, and biological knowledge base design. *Prerequisite: Pharm 201 or consent of instructor.*

CSE 257A. Bioinformatics II: Sequence and Structure Analysis—Methods and Applications (4)
Introduction to methods for sequence analysis. Applications to genome and proteome sequences. Protein Structure, sequence-structure analysis.

CSE 257B. Algorithms in Computational Biology (4)
Renumbered to CSE 206B (see above).

CSE 258A. Connectionists Natural Language Processing (4)
(Formerly CSE 281P.) This course will explore connectionist (or parallel distributed processing) models and their relation to cognitive processes. The course will cover various learning algorithms and the application of the paradigm to models of language processing, memory, sequential processes, and vision. (S/U grades permitted.) *Prerequisite: CSE 250B or equivalent experience.*

CSE 259. Seminar in Artificial Intelligence (1)
A weekly meeting featuring local (and occasional external) speakers discussing their current research in Artificial Intelligence Neural Networks, and Genetic Algorithms. (S/U grades only.) *Prerequisite: none.*

CSE 260. Parallel Computation (4)
(Formerly CSE 274A.) This course provides an overview of parallel hardware, algorithms, models, and software. Topics include Flynn's taxonomy, interconnection networks, memory organization, a survey of commercially available multiprocessors, parallel algorithm paradigms and complexity criteria, parallel programming environments and tools for parallel debugging, language specification, mapping, performance, etc. *Prerequisite: graduate standing or consent of instructor.*

CSE 262. System Support for Applications of Parallel Computation (4)
This course will explore design of software support for applications of parallel computation. Topics include: programming languages, run time support, portability, and load balancing. The course will terminate in a project. *Prerequisite: consent of instructor.*

CSE 268A. Topics in Parallel Computation (4)
(Formerly CSE 281Y.) Current topics of interest in parallel computation will be discussed such as heterogeneous computing, advanced topics in parallel programming environments, parallel programming models, performance criteria, etc. (S/U grades permitted.) *Prerequisite: graduate standing or consent of instructor.*

CSE 268C. Topics in High-Performance Programming (4)
A systematic approach to the design, writing, and tuning of programs to sustain near-peak performance with particular emphasis on RISC processors and massively parallel computers. A project will involve measuring and improving the performance of a computational kernel. *Prerequisite: CSE 141 or consent of instructor.*

CSE 269. Seminar in Parallel Computation (1-4)
A seminar course in which topics of special interest in parallel computation will be presented by staff members and graduate students under faculty direction. Topics vary from quarter to quarter. May be repeated for credit. *Prerequisite: consent of instructor.* (S/U grades only.)

CSE 271. User Interface Design: Social and Technical Issues (4)
Web technologies (HTML, Java, JavaScript, etc.) can quickly build superb new systems, as well as phenomenally ugly systems that still fully meet their performance and functional requirements. This course explores interface usability and representation issues, with some focus on hypermedia and cooperative work. *Prerequisites: CSE 20, CSE 100, or equivalent.*

CSE 275. Social Aspects of Technology and Science (4)
Note: CSE 275 is pending CEP approval. Explores approaches to the sociology of technology and science, especially information technology. Topics include requirements engineering, actor-network theory, post-modernism, the Web, user interface design, and public policy. *Prerequisites: CSE 8B or CSE 11, and background in the humanities.*

CSE 290. Seminar in Computer Science and Engineering (1-4)
(Formerly CSE 280A.) A seminar course in which topics of special interest in computer science and engineering will be presented by staff members and graduate students under faculty direction. Topics vary from quarter to quarter. May be repeated for credit. (S/U grades only.) *Prerequisite: consent of instructor.* (Offered as faculty resources permit.)

CSE 291. Topics in Computer Science and Engineering (1-8)
(Formerly CSE 281A). Topics of interest in computer science and engineering. Topics may vary from quarter to quarter. May be repeated for credit with the consent of instructor. (S/U grades permitted.) *Prerequisite: consent of instructor.* (Offered as faculty resources permit.)

CSE 292. Faculty Research Seminar (1)
(Formerly CSE 282.) Computer science and engineering faculty will present one hour seminars of the current research work in their areas of interest. *Prerequisite: CSE graduate status.*

CSE 293. Special Project in Computer Science and Engineering (1-8)
(Formerly CSE 269.) The student will conceive, design, and execute a project in computer science under the direction of a faculty member. The project will typically include a large programming or hardware design task, but other types of projects are possible. One-six units may be repeated to a total of nine units. *Prerequisite: CSE graduate student status.* (S/U grades only.)

CSE 294. Research Meeting in CSE (2)
Advanced study and analysis of active research in computer science and computer engineering.

Return To Contents

Discussion of current research and literature in the research specialty of the staff member teaching the course. *Prerequisite: consent of instructor.*

CSE 298. Independent Study (1-16)
Open to properly qualified graduate students who wish to pursue a problem through advanced study under the direction of a member of the staff. (S/U grades only.) *Prerequisite: consent of instructor.*

CSE 299. Research (1-16)
Research. *Prerequisite: consent of faculty.*

CSE 500. Teaching Assistantship (2-4)
A course in which teaching assistants are aided in learning proper teaching methods by means of supervision of their work by the faculty: handling of discussions, preparation and grading of examinations and other written exercises, and student relations. May be used to meet teaching experience requirement for candidates for the Ph.D. degree. Number of units for credit depends on number of hours devoted to class or section assistance. *Prerequisites: graduate standing and consent of instructor.*

CSE 501. Teaching Assistantship (2-4)
Renumbered to CSE 500 (see above).

CSE 599. Teaching Methods in Computer Science (2)
Training in teaching methods in the field of computer science. This course examines theoretical and practical communication and teaching techniques particularly appropriate to computer science. *Prerequisite: consent of faculty.*