tion techniques. Nature of turbulence. Reynolds equations. Introduction to turbulence modeling. Offered in alternate years.—II. (B) Bondi.
280A. Nonlinear Finite Elements for Elastic-Plastic Problems (4) Lecture—4 hours. Prerequisite: consent of instructor. State of the art finite element methods and tools for elastic-plastic problems, including computational techniques based on the finite element method and the theory of elasto-plasticity. Offered in alternate years.—III. (I) Jeremic
280B. Nonlinear Dynamic Finite Elements (4) Lecture—4 hours. Prerequisite: consent of instructor. State of the art computational methods and tools for analyzing linear and nonlinear dynamics problems. Offered in Jeremic.
281A. Advanced Soil Mechanics (4) Lecture—4 hours. Prerequisite: course 171. Soil investigation and site characterization within the context of slope stability analysis.—II. (I) Dejong
282. Pavement Design and Rehabilitation (4) Lecture—4 hours. Prerequisite: course 179 or consent of instructor. Advanced pavement design and structural/functional condition evaluation for concrete and asphalt pavements. Highways, airfields, port facilities; new facilities, rehabilitation, reconstruction. Mechanistic-empirical procedures, materials, climate and traffic characterization. Use of current design methods; recent developments and research. Offered in alternate years.—III. (I) Harvey
283. Physical-Chemical Aspects of Soil Behavior (4) Lecture—3 hours; laboratory—3 hours. Prerequisite: course 171. Study of the geotechnical behavior of soils considering transportation, transport, mineralogy, soil-fluid-electrolyte systems, surface tension, particle mechanics, shape, fabric, and structure. Laboratories demonstrate effects of fundamental interparticle forces (contact, Van Der Waals, capillarity and chemical); offered in alternate years.—I. (I) Kutter
286. Advanced Foundation Design (4) Lecture—4 hours. Prerequisite: course 173. Design and analysis of pile and pier foundations, including seismic effects; deep excavation systems; tie-back, nailing, and anchor systems; coffer dams; loads on buried conduits; ground modification techniques; and other related topics.—III. (I) Dejong
287. Geotechnical Earthquake Engineering (4) Lecture—4 hours. Prerequisite: courses 138 and 281A. Characteristics and estimation of earthquake ground motion, soil site amplification and local site response; liquefaction potential and remediation; residual strength and stability considerations; ground deformations; dynamic soil-structure interaction.—III. (I) Dejong
288. Earth and Rockfill Dams (4) Lecture—4 hours. Prerequisite: courses 281A and 281B (may be taken concurrently). Site selection; design considerations; layout; seismic effects including coupled motion; soil movements; construction; environmental considerations, instrumentation; maintenance remediation and retrofit of existing dams. Offered in alternate years.—II. (I) Jeremic
289A-I. Selected Topics in Civil Engineering (1-5) Lecture, laboratory, or combination. Prerequisite: consent of instructor. Directed group study of special topics with separate sections in (A) Environmental Engineering; (B) Hydraulic and Hydrologic Engineering; (C) Engineering Planning, (D) Geotechnical Engineering; (E) Structural Engineering; (F) Structural Mechanics; (G) Transportation Engineering; (H) Transportation Planning; (I) Water Resources Engineering. [I] Water Resources Planning. May be repeated for credit. —I, II, III, (I, II, III)
290. Seminar (1) Seminar—1 hour. Discussion of current graduate research, and guest lectures on recent advances. Oral presentation of individual study. Consent required of graduate advisor. [S/U grading only]—I, II, III, (I, II, III)
290C. Graduate Research Conference (1) Lecture—4 hours. Prerequisite: consent of instructor. [S/U grading only]
299. Research (1-12) [S/U grading only]
Professional
390. The Teaching of Civil Engineering (1) Discussion—1 hour. Prerequisite: meet qualifications for teaching assistant and/or associate-in in Civil Engineering. Participation as teaching assistant or associate-in in a designated engineering course. Methods of leading discussion groups or laboratory sections, writing and grading quizzes, use of laboratory equipment and laboratory report. May be repeated for total of 9 units. [S/U grading only]—I, II, III, (I, II, III)
398. Group Study (1-5) Prerequisite: consent of instructor. [S/U grading only]
Engineering: Computer Science

[College of Engineering] Nina Amenta, Ph.D., Chairperson of the Department Department Office. 2063 Kemper Hall 5307/29270064; http://www.cs.ucdavis.edu

Faculty
Nina Amenta, Ph.D., Professor Zhaojun Bai, Ph.D., Professor Matthew Bishop, Ph.D., Professor Hao Chen, Ph.D., Associate Professor Ian Davidson, Ph.D., Associate Professor Premkumar T. Devanbu, Ph.D., Professor Matthew K. Farrens, Ph.D., Professor Vladimir Filkov, Ph.D, Associate Professor Matthew Franklin, Ph.D., Professor Dipak Ghosal, Ph.D., Professor Todd J. Green, Ph.D., Assistant Professor Daniel Gufstein, Ph.D., Professor Francios Gygi, Ph.D., Professor Bernd Hamann, Ph.D., Professor Kenneth I. Joy, Ph.D., Professor Academic Senate Distinguished Teaching Award Patrick Keglarıyla, Ph.D., Associate Professor Karl Levitt, Ph.D., Professor Xi Lin, Ph.D., Associate Professor Bertram Ludescher, Ph.D., Professor Kwan-Liu Ma, Ph.D., Professor Charles U. Martel, Ph.D., Professor Norman S. Matloff, Ph.D., Professor Nelson Max, Ph.D., Professor Prasant Mohapatra, Ph.D., Professor Biswanath Mukherjee, Ph.D., Professor Distinguished Graduate Mentoring Award Michael Neff, Ph.D., Assistant Professor Ronald A. Olsson, Ph.D., Professor Academic Senate Distinguished Teaching Award Raju Pandey, Ph.D., Associate Professor Phillip Raygoza, Ph.D., Professor Zhendong Su, Ph.D., Associate Professor Ilias Tagkopoulos, Ph.D., Assistant Professor S. Felix Wu, Ph.D., Professor Emeriti Faculty Lawrence T. Kou, Ph.D., Professor Emeritus Peter Linz, Ph.D., Professor Emeritus Rolf F. Walters, Ph.D., Professor Emeritus Academic Senate Distinguished Teaching Award

Affiliated Faculty Sean Davis, M.S., Lecturer Christopher Nitta, Ph.D Assistant Adjunct Professor Sean Peisert, Ph.D, Assistant Adjunct Professor Owen Carmichael, Ph.D, Assistant Adjunct Professor Gunther Webber, Ph.D, Assistant Adjunct Professor The Computer Science and Engineering Program The Department of Computer Science administers two curricula: Computer Science and Engineering in the College of Engineering, and Computer Science in the College of Letters and Science. It also administers two minors: Computer Science in the College of Letters and Science, and Computational Biology in the College of Engineering. For information on the Computer Science curriculum and minor; see Computer Science, on page 216. The Computer Science Engineering major (below) prepares students to do further work in hardware, software, theory, or electronics, either in industry or in postgraduate study. The primary differences between the Computer Science Engineering and the Computer Science majors are the extent of course work covering hardware and the flexibility of the curriculum. The Computer Science Engineering major develops a solid understanding of the entire machine, including hands-on experience with its hardware components. The Computer Science major has some course work on hardware, at the digital-design level, on simulators. The Computer Science Engineering minor has fewer free electives. The CS major’s more generous electives make it easier to complete a minor or double major. A key theme of the Computer Science Engineering curriculum is the hardware/software interaction, a theme reflected in the courses required and the orientation of the courses themselves. The Computer Science and Engineering major provides students with a solid background in mathematics, physics, chemistry, and electronic circuits and systems, all supporting the computer hardware and computer software courses that constitute the focus of the curriculum. Mission. The University of California, Davis, is, first and foremost, an institution of learning and teaching, committed to serving the needs of society. The Department of Computer Science and Engineering contributes to the mission in three ways. First, its undergraduate and graduate education programs seek to educate students in the fundamental principles of computer science and the skills needed to solve the complex technological problems of modern society. The breadth of coursework provides a framework for life-long learning and an appreciation for multidisciplinary activity. Second, through its research programs, the department contributes to the development and progress of computer science, and software and information technology, to provide innovative, creative solutions for societal needs. Finally, the department disseminates its research to enhance collaborations with the public sector, further interdisciplinary interests that benefit society, and
educate the public through publications, public service, and professional activities.

**Department Objectives.** Teaching—We seek to provide students with a thorough understanding of the key principles and practices of computing, which include a strong theoretical background in mathematics, basic sciences, and engineering fundamentals. We desire to give students an ability to apply their knowledge to practical problems. We endeavor to provide students with sufficient breadth to work creatively and productively in multidisciplinary work teams. A broad breadth, in its broadest context, will form the basis for an appreciation and interest in lifelong learning. We provide students with the opportunities to design and conduct experiments, and to collect and analyze data in core, as well as more specialized, areas of computer science. We provide students with breadth in the humanities and social sciences so they can learn to communicate effectively, understand professional and ethical issues in society, and appreciate the interrelatedness between computing and society. We educate graduate students to be our next generation of teachers or leaders in industry, or to pursue meaningful, creative research in industry, government, or academia. Research—We develop and maintain research programs that produce fundamental scientific advances, as well as useful technological innovations, while simultaneously training the next generation of researchers and leaders in the field of computer science.

**Objectives.** We train graduates to practice computer science and engineering in a broad range of industries. We encourage graduates for graduate education or other professional degrees; we give students an understanding of computer software and hardware systems, and both theoretical and experimental approaches to problem solving; we ready graduates for lifelong learning; and we encourage graduates to contribute to their profession and society.

**Integrated Degree Program.** An integrated B.S./M.S. plan in Computer Science allows Davis students in Computer Science, Computer Science Engineering, or Computer Engineering to complete a master’s degree in Computer Science in one year. Formal course work for the master’s degree is reduced by six units for these students. Students can begin graduate studies immediately after completing their B.S. degree. More information is available in the graduate section of the College of Engineering Bulletin, or at http://www.cs.ucdavis.edu/graduate/bsms.html.

**Computer Science and Engineering Undergraduate Program**

The Computer Science and Engineering program is accredited by the Engineering Accreditation Commission of ABET, see http://www.abet.org.

Students are encouraged to adhere carefully to all prerequisite requirements. The instructor is authorized to drop students from a course for which stated prerequisites have not been completed.

### Lower Division Required Courses

| Course Code | Title | UNITS |普
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Math 21A, 21B, 21C, 21D</td>
<td>Mathematics</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Math 22A or 67</td>
<td>Mathematics</td>
<td>3-4</td>
<td></td>
</tr>
<tr>
<td>Phys. 9A-9B, 9C-9D</td>
<td>Physics</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>Chem. 2A</td>
<td>Chemistry</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Comp. Science 20, 30, 40</td>
<td>Computer Science</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>Comp. Science 50</td>
<td>Computer Science</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Electrical and Computer Engineering 70</td>
<td>Electrical and Computer Engineering</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Engin. 17</td>
<td>Engineering</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>English 3 or University Writing Program 1</td>
<td>English</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>1V, 1Y or Comparative Literature 1, 2, 3</td>
<td>Comparative Literature</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Native Americans 1, 2</td>
<td>Native American Studies</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Communication 1</td>
<td>Communication</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

### Upper Division Requirements: Upper Division Required Courses

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Title</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Science Engineering 132</td>
<td>Computer Science Engineering</td>
<td>4</td>
</tr>
<tr>
<td>Electrical and Computer Engineering 100</td>
<td>Electrical and Computer Engineering</td>
<td>4</td>
</tr>
<tr>
<td>Comp. Science electives</td>
<td>Computer Science</td>
<td>12</td>
</tr>
<tr>
<td>Math 102</td>
<td>Mathematics</td>
<td>4</td>
</tr>
<tr>
<td>University Writing Program 101</td>
<td>University Writing Program</td>
<td>4</td>
</tr>
</tbody>
</table>

**Upper Division Composition Requirement**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Title</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comp. Science Engineering 120</td>
<td>Computer Science Engineering</td>
<td>4</td>
</tr>
<tr>
<td>Comp. Science Engineering 122A</td>
<td>Computer Science Engineering</td>
<td>4</td>
</tr>
</tbody>
</table>

**Computational Biology Minor**

The minor in Computational Biology will provide students with engineering, physical science or biological science majors the foundations necessary to build efficient computational models and algorithms, use state-of-the-art software for scientific analysis, and create scalable software environments for biological and biotechnological applications.

Students must take a total of 20 upper division units, with two required courses and 12 units of upper division electives, as specified below. A minimum GPA of 2.000 is required for coursework in the minor. Students should note that most of the courses listed below have lower division prerequisites. In particular, required course Computer Science Engineering 122A has a prerequisite chain of Computer Science Engineering 20, 30, and 40.

**Minor Requirements**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Title</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comp. Science 122A, 124</td>
<td>Electives</td>
<td>12</td>
</tr>
</tbody>
</table>

At least one biology course from the following: Molecular & Cellular Biology 121 A, 121 B, 122 A, 122 B, Evolution and Ecology 100, 102, 104, 131; Biological Sciences 101, 104, 122

At least one computational or statistics course from Computer Science Engineering 130, 132, 140, 145, 156, 158, 160, 165A, 166, 170, 177; Evolution and Ecology 175, Statistics 130A, 141; Biotechnology 150; Biological Sciences 132

At least one computational biology and bioinformatics course from the following: Computer Science Engineering 129; Biological Sciences 132, Biomedical Engineering 117, Evolution and Ecology 175, Biotechnology 150

### Minors

<table>
<thead>
<tr>
<th>Minor Advisor</th>
<th>Title</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. L. Avellar, Vladimir Filkov, Dan Gusfield, Patrice Koehl, Bertram Laueaesser, Ilia Tagkopoulos</td>
<td>Minor Advisers</td>
<td></td>
</tr>
</tbody>
</table>

**Courses in Engineering: Computer Science (ECS)**

**Lower Division**

### 10. Introduction to Programming (4)

**Lecture—3 hours; discussion—1 hour. Prerequisite:** two years of high school algebra. A hands-on introduction to computation, through programming and problem solving. GE credit: SciEng | QL, SE, SL—II, III, IV, V, VI, VII, VIII, Amento, Gettler, Ludaescher

### 12. Introduction to Media Computation (4)

**Lecture—3 hours; discussion/labatory—1 hour. Prerequisite:** to key computational ideas necessary to understand and produce media. Fundamentals of programming are covered as well as analysis of how media are represented and transmitted in digital form. Aimed primarily at non-computer science students. [Some course as Cinema and Techno-cultural Studies 012] GE credit: ArtHum or SciEng | AH or SE, VL—II, III, IV, V, VI, VII, Amento

### 13. Introduction to Computers (4)

**Lecture—3 hours; laboratory—3 hours. Not open for credit to students who have completed course 30. Computer uses in modern society. Emphasis on uses in nontechnological fields. Includes word processing, spreadsheets, web-page creation, electronic programming, basic computer organization, and use of the Internet, the uses of computers and their influence on society. Course not intended for CS or CSE majors. Only two units of credit allowed to students who have completed Plant Science 21. GE credit: SciEng, Wrt | QL, SE, WE—II, III, IV, V, VI, VII, Amento

### 20. Discrete Mathematics for Computer Science (4)

**Lecture—3 hours; discussion—1 hour. Prerequisite:** grade of C- or better in Mathematics 16A, 17A or 21A. Discrete mathematics of particular utility to computer science. Proofs by induction. Propositional and first-order logic. Sets, relations, equivalence relations, Big-O and related notations. Recursion and solutions of recurrence relations. Combinatorics. Probability on finite probability spaces. Graph theory. GE credit: SciEng | QL, SE—II, III, IV, V, VI, VII, Amento, Tagkopoulos

### 30. Programming and Problem Solving (4)

**Lecture—3 hours; discussion—1 hour. Prerequisite:** Mathematics 16A or 21A may be taken concurrently; prior experience with basic programming concepts (variables, loops, conditional statements) recommended. Introduction to computers and computer programming, algorithm design, and debugging. Elements of good programming style. Use of basic UNIX tools. GE credit: SciEng | QL, SE, SL—II, III, IV, V, VI, VII, Amento

### 40. Software Development and Object-Oriented Programming (4)

**Lecture—3 hours; discussion—1 hour. Prerequisite:** course 30 or the equivalent with a grade of C- or better. Elements of program design, style, documentation, efficiency. Methods for debugging and verification. Operating system tools. Principles and use of object-oriented programming in C++. Design of object-oriented programs. Introduction to I/O devices and programming. Only one unit of credit allowed for students who have taken Electrical and Computer Engineering 70. GE credit: SciEng | SE, SL—II, III, IV, V, VI, VII, Amento

### 50. Computer Organization and Machine-Dependent Programming (4)

**Lecture—3 hours; discussion—1 hour. Prerequisite:** course 40. Comparative study of different hardware architectures via programming in the assembly languages of various machines. Role of system software in producing an abstract machine. Introduction to I/O devices and programming. Only one unit of credit allowed for students who have taken Electrical and Computer Engineering 70. GE credit: SciEng | SE, SL—II, III, IV, V, VI, VII, Amento

### 60. Data Structures and Programming (4)

**Lecture—3 hours; discussion—1 hour. Prerequisite:** courses 20, 40 (C++ and UNIX); grade of C- or better in each course. Design and analysis of data structures for a variety of applications. Trees, heaps, searching, sorting, hashing, graphs. Extensive programming. GE credit: SciEng | QL, SE, SL—II, III, IV, V, VI, VII, Amento

### 89A-L. Special Topics in Computer Science (1-5)

Lecture, laboratory or combination. Prerequisite: consent of instructor. Special topics in (A) Computer Science Theory; (B) Architecture; (C) Programming Languages and Compilers; (D) Operating Systems; (E) Software Engineering; (F) Databases; (G) Artificial Intelligence; (H) Computer Graphics; (I) Network; (J) Computer-Aided Design; (K) Scientific Computing; (L) Domains of Application.
92. Internship in Computer Science (1-5)
Internship. Prerequisite: lower division standing; project approval prior to period of internship. Supervised work experience in computer science may be repeated for credit (P/NP grading only).

98. Directed Group Study (1-5) 
[P/NP grading only]

99. Special Study for Lower Division Students (1-5) 
[P/NP grading only]

Upper Division

120. Theory of Computation (4) 

122A. Algorithm Design and Analysis (4) 

122B. Algorithm Design and Analysis (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 122A. Theory and practice of hard problems, and problems with complex algorithm solutions. NP-completeness, approximation algorithms, randomized algorithms, dynamic programming and branch and bound. Theoretical analysis, implementation and practical evaluations. Examples from parallel, string, graph, and geometric algorithms. GE credit: SciEng | QL, SE—I, (I) Gushel, Martel, Rogaway

124. Theory and Practice of Bioinformatics (4) 
Lecture—3 hours; laboratory—1 hour. Prerequisite: course 10 or 30 or Engineering 6; Statistics 12 or 13 or 32 or 100 or 131A or Mathematics 135A; Biological Science 1A or Molecular and Cellular Biology 10. Fundamental biological, mathematical and algorithmic models underlying bioinformatics and systems biology; sequence analysis, database search, gene registration, clustering and classification, functional gene networks, regulatory network inference, phylogenetic trees, applications of common bioinformatics tools in molecular biology and genetics. GE credit: SciEng | SE—III, (III) Gushel, Filakov, Tagkopoulos

127. Cryptography (4) 
Lecture—3 hours; discussion—1 hour. Introduction to the theory and practice of cryptographic techniques used in computer security. Encryption (secret key and public key), message authentication, digital signatures, entity authentication, key distribution, and other cryptographic protocols. The social context of cryptography. GE credit: SciEng | QL, SE—SL—Franklin, Gushel

129. Computational Structural Bioinformatics (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: college-level programming course; Biological Science 1A or Molecular and Cellular Biology 10. Fundamental biological, chemical and algorithmic models underlying computational structural biology; protein structure and nucleic acids structure; comparison of protein structures; protein structure prediction, molecular simulations; databases and online services in computational structural biology. GE credit SciEng | SE—I, II, Koehler

130. Scientific Computation (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 30 or 60, Mathematics 23A or Mathematics 67. Matrix-based approach using MATLAB for floating-point arithmetic, error analysis, data interpolation, least squares data fitting, quadrature, zeros, optimization and matrix eigenvalues and singular values. Parallel matrix operations and essential matrix factorizations. GE credit: SciEng | SE—III, (III) Bai, Hamann, Joy

132. Probability and Statistical Modeling for Computer Science (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 40, course 50 or Engineering Electrical and Computer 70; Mathematics 21C; Mathematics 22A or Mathematics 67. Univariate and multivariate distributions. Estimation, hypothesis testing, Markov/Hidden Markov models. Applications to data mining, networks, security, software engineering and bioinformatics. GE credit: SciEng | QL, SE—II, (II) Davidović, Ghosal

140A. Programming Languages (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 50 or Electrical Computer Engineering 70, course 60. Syntactic definition of programming languages. Introduction to programming language features including variables, data types, data abstraction, object-orientedness, scoping, parameter disciplines, exception handling. Non-imperative programming languages. Study of several high-level programming languages. GE credit: SciEng | SE—I, II, (II) Olsson, Pandey, Su

140B. Programming Languages (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 140A. Continuation of programming language principles. Further study of programming language paradigms such as functional and logic; additional programming language paradigms such as concurrent (parallel); key implementation issues for those paradigms; and programming language semantics. Offered in alternate years. GE credit: SciEng | SE—I, II, (II) Levit, Olsson, Pandey

142. Compilers (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 20, 140A; course 120 recommended. Principles and techniques of lexical analysis, parsing, semantic analysis, code generation, and code optimization. Implementation of compilers. GE credit: SciEng | SE—I, II, (II) Pandey, Su

145. Scripting Languages and Their Applications (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: programming skill at the level of course 60. Goals and philosophy of scripting languages, with Python and R as prime examples. Applications include networking, data analysis and display, and graphical user interfaces. Offered in alternate years. GE credit: SciEng | SE—I, II, (II) Pandey, Maloff

150. Operating Systems and System Programming (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 50 or Electrical and Computer Engineering 70. Basic concepts of operating systems and system programming. Processes and interprocess communication/synchronization; virtual memory, program loading and linking; file and I/O subsystems; utility programs. Study of a real operating system. GE credit: SciEng | SE—I, II, III, (II, III) Levit, Maloff, Olsson, Wu

152A. Computer Architecture (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 60; course 132 or Electrical and Computer Engineering 161 or Mathematics 135A or Statistics 131A, or Statistics 120 or Statistics 22. Overview of computer networks; Internet protocol suite, computer networking applications and protocols, transport-layer protocols, network architectures, Internet Protocol (IP), routing, link-layer protocols, local area and wireless networks; security and access control, physical aspects of data transmission, and network performance analysis. Only 2 units of credit for students who have taken course 157. (Same course as Electrical and Computer Engineering 173A.) GE credit: SciEng | SE—I, II, III, (II, III) Chuah, Ghosal, Liu, Matloff, Mohapatra, Mukherjee

152B. Computer Networks (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 152A or Electrical and Computer Engineering 173A. TCP/IP protocol suite, computer networking applications, client-server architectures, application-layer protocols, transport-layer protocols, transport-layer interfaces, sockets, network programming, remote procedure calls, and network management. GE credit: SciEng | SE—I, II, III, (II, III) Ghosal, Matloff, Mohapatra, Mukherjee

152C. Advanced Topics in Computer Networks (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 152A or Electrical and Computer Engineering 173A. Advanced topics in computer networks, wireless networks, multimedia networking, traffic analysis and modeling, network design and management, network programming, and performance analysis, and design projects in communication networks. (Same course as Electrical and Computer Engineering 173B.) Offered in alternate years. GE credit: SciEng | SE—I, II, III, (II, III) Chuah, Liu, Mukherjee, van der Schaar

153. Computer Security (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: courses 150 and 152A. Principles, mechanisms, and implementation of computer security and data protection. Policy, encryption and authentication, access control, and integrity models and mechanisms; network security; secure systems; programming and vulnerabilities and attack vectors for existing operating system. Not open for credit to students who have completed course 155. GE credit: SciEng | SE—I, II, III, (III) Bishop, Chen

154A. Computer Architecture (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 50 or Electrical and Computer Engineering 70. Introduction to digital design. Interfacing of devices for I/O, memory and memory management. Input/output programming, via wait loops, hardware interrupts and calls to operating system services. Hardware support for operating systems software. GE credit: SciEng | SE—I, II, III, (II, III) Matloff, Pandey, Su

158. Programming on Parallel Architectures (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 154A or both Electrical and Computer Engineering 170 and Electrical and Computer Engineering 180A. Hardwired and software CPU design. Memory hierarchies. Uniprocessor performance analysis under varying program mixes. Introduction to pipelining and multiprocessors. GE credit: SciEng | SE—I, II, III, (II, III) Pandey, Su

158. Programming on Parallel Architectures (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: courses 150 and 154B recommended. Techniques for software development using the shared-memory and message-passing paradigms, on parallel architectures and networks of workstations. Lacks, barriers, and other techniques for synchronization. Introduction to parallel algorithms. GE credit: SciEng | SE—I, II, III, (II, III) Chiang, Ferrans, Ma, Matloff, Pandey, Su

160. Software Engineering (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 140A. Requirements specification, design, implementation, testing, and verification of large software systems. Study and use of software engineering methodologies. Team programming. GE credit: SciEng | SE—I, II, III, (II, III) Devanbu, Levitt

163. Information Interfaces (4) 
Lecture—3 hours; discussion—1 hour. Prerequisite: course 60. Art and science of information visualization and interfaces for information systems. Design principles of human-computer interaction. Visual display and navigation of nonspatial and higher dimensional data. Implementations, performance issues,
tradeoffs, and evaluation of interactive information systems. GE credit: SciEng | SE, VL.—III. (Ill.) Amenta, Ma

165A. Database Systems (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 60. Database modeling and design (ER model, relational model), relational algebra, query languages (SQL), file and index structures, query processing, transaction management. GE credit: SciEng | SE.—II. (II.) Ludascher

165B. Database Systems (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 165A. Data modeling (object-relational, graph-based, spatiotemporal models). Querying semistructured data (XML). Database theory (normalization, integration, provenance). Database programming (stored procedures, embedded SQL, web programming). Advanced topics (data warehousing, parallel data processing). GE credit: SciEng | SE.—III. (Ill.) Ludascher

170. Introduction to Artificial Intelligence (4)

171. Machine Learning (4)
Lecture—3 hours; discussion—1 hour. Introduction to machine learning and unsupervised learning, including classification, dimensionality reduction, regression and clustering using modern machine learning methods. Applications of machine learning in healthcare. GE credit: SciEng | SE.—II. (II.) Davidson, Matloff, Tagkopoulos

173. Image Processing and Analysis (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 60; Mathematics 67 or C or better in Mathematics 22A. Techniques for automated extraction of high-level information from images generated by cameras, three-dimensional surface sensors, and medical devices. Typical applications include detection of objects in various types of images and describing populations of biological specimens appearing in medical imagery. GE credit: SciEng | SE.—II. (II.) Amenta

175. Computer Graphics (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 60; Mathematics 22A or Mathematics 67. Principles of computer graphics, with a focus on interactive systems. Current graphics hardware, elements and applications in three- and higher-dimensional space, geometric transformations, camera models and interaction, graphics system design, standard graphics APIs, individual projects. GE credit: SciEng | SE.—II. (II.) Amenta, Hamann, Jay

177. Scientific Visualization (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 175. Computer graphics techniques for generating images of various types of measured or computer-simulated data. Typical applications for these graphics techniques include study of air flows around car bodies, medical data, and molecular structures. GE credit: SciEng | SE, VL.—II. (II.) Hamann, Joy, Max

180. Geometric Modeling (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 175. Interactive graphics techniques for defining and manipulating geometric shapes used in computer-aided design, aircraft design, and architectural design. GE credit: SciEng | SE, VL.—I. (I.) Hamann, Joy, Max

188. Ethics in an Age of Technology (4)

189A-N. Special Topics in Computer Science (1-5)

190C. Research Group Conferences in Computer Science (1)
Discussion—1 hour. Prerequisite: upper division standing in Computer Science and Engineering; consent of instructor. Research group conferences. May be repeated for credit. (P/NP grading only.)—I, II, III, (I, II, III)

190X. Senior Seminar (2)
Seminar—2 hours. Prerequisite: senior standing. Examination of a special topic in a small group setting.—I, II, III, (I, II, III)

192. Internship in Computer Science (1-5)
Internship. Prerequisite: completion of a minimum of 84 units; project approval prior to period of internship. Supervised work experience in computer science. May be repeated for credit. (P/NP grading only.)

193A. Senior Design Project (2)
Lecture—1 hour; laboratory—3 hours. Prerequisite: course 160 recommended (may be concurrent) or consent of instructor. Open to Computer Science or Computer Science and Engineering seniors. Team design project involving analysis, design, implementation and evaluation of a large-scale problem involving computer and computational systems. The project is supervised by a faculty member. Students must take course 193A and 193B to receive credit. (Deferred grading only, pending completion of sequence.) GE credit: SciEng | SE.—II, III, (II, III.) Davidson, Joy, Mohapatra

193B. Senior Design Project (2)
Lecture—1 hour; laboratory—3 hours. Prerequisite: IP grade in course 193A. Team design project involving analysis, design, implementation and evaluation of a large-scale problem involving computer and computational systems. The project is supervised by a faculty member. Students must take course 193A and 193B to receive credit. (Deferred grading only, pending completion of sequence.) GE credit: SciEng | SE.—II, III, (II, III.) Davidson, Joy

197T. Tutoring in Computer Science (1-3)
Discussion—1 hour; laboratory/discussion—3-6 hours. Prerequisite: consent of instructor. Restricted to upper-division standing. Tutoring in computer science courses, especially introductory courses. (P/NP grading only.)—I, II, III, (I, II, III)

198. Directed Group Study (1-5)
Prerequisite: consent of instructor. (P/NP grading only)

199. Special Study for Advanced Undergraduates (1-5)
(P/NP grading only)

The Graduate Program in Computer Science

Graduate

210A. Advanced Computer Architecture (4)
Lecture—3 hours; term paper. Prerequisite: course 154B or Electrical and Computer Engineering 170; course 150: Modern research topics and methods in computer architecture. Design implications of memory latency and bandwidth limitations. Performance enhancement via within-processor and between-processor parallelism. Term project involving student-proposed extensions/modifications of work in the research literature. Not open for credit to students who have completed course 250A.—I. Farrens

218B. High-Performance Uniprocessoring (4)
Lecture—3 hours; term paper. Prerequisite: course 191A. Maximizing uniprocessor performance. Barriers to high performance; solutions to the problems; historical and current processor designs. Not open for credit to students who have completed course 250B.—II. Farrens

211C. Parallel Architectures (4)
Lecture—3 hours; project—1 hour. Prerequisite: course 210A. Evolution of parallel architectures from special-purpose machines to commodity. Emphasis on recent machines and applications that drive them. Not open for credit to students who have completed course 250C.—III

203. Novel Computing Technologies (4)
Lecture—3 hours; project—1 hour. Prerequisite: course 210A. Novel computing technologies that could revolutionize computer architecture. Quantum computing technologies, including algorithms, devices, and fault tolerance. A survey of unconventional technologies including nanoscale electronics, MEMS devices, biological devices, and nanotechnology. Offered in alternate years.—II

220. Theory of Computation (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 120, 122A. Time and space complexity classes. Reductions, completeness, and the role of randomness. Logic and undecidability.—III. RagHAVEN

221. Computational Methods in Systems and Synthetic Biology (4)

222A. Design and Analysis of Algorithms (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 212A; Statistics 131A recommended. Techniques for designing efficient algorithms, analyzing their complexity and applicability to a broad range of applications. Methods for recognizing and dealing with difficult problems.—I, II, (II, II.) Amenta, Franklin, Gusfield, Martin, RagHAVEN

222B. Advanced Design and Analysis of Algorithms (4)
Lecture—3 hours; project—1 hour. Prerequisite: course 222A. Advanced topics in complexity theory. Problem classification. The classes P, NP, P-space, coNP, Matching and network flow algorithms. Matrix multiplication. Approximation algorithms.—III. (III.) Gusfield, Franklin, Martel, RagHAVEN

223. Parallel Algorithms (4)
Lecture/discussion—3 hours; project—1 hour. Prerequisite: course 222A. Design and analysis of parallel computer systems including PRAMs, loosely coupled systems and interconnection networks. Parallel algorithms for classical problems and general techniques for their design and analysis. Proving lower bounds on parallel computation in several settings.—II. (II.) Martel

224. String Algorithms and Applications in Computational Biology (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 222A. Algorithms that operate on strings. Pattern matching, sets of patterns, regular expression pattern matching, suffix trees and applications, inex-
act similarity, parametric sequence alignment, applications to DNA sequencing and protein database searches. Offered in alternate years.—Ill. Gusfield, Filkov


226. Computational Geometry (4) Lecture—3 hours; discussion—1 hour. Prerequisite: course 175, 222A. Mathematics of unstructured data. Algorithms for data structures such as Voronoi diagrams, octrees, and arrangements. Applications in computer graphics, concentrating on problems in three-dimensions. Offered in alternate years.—III. Amenta, Max

227. Modern Cryptography (4) Lecture—3 hours; discussion—1 hour. Prerequisite: course 225, 222A. Modern cryptography as a discipline emphasizing formal definitions and proofs of security. One-way functions, pseudo-randomness, encryption, digital signatures, zero-knowledge, secure two-party protocols. Offered in alternate years.—I, III. (I, III.) Filkov

228. Cryptography for E-Commerce (4) Lecture—3 hours; discussion—1 hour. Prerequisite: course 222A. Cryptographic primitives and protocols of importance to e-commerce, present and future. Distribution mechanisms such as payment mechanisms, pricing mechanisms, anonymity and privacy mechanisms, fair exchange mechanisms. Offered in alternate years.—II. Franklin

229. Advanced Computational Structural Bioinformatics (4) Lecture—3 hours; discussion—1 hour. Prerequisite: graduate standing. Algorithmic problems in structural biology; protein structure classification; protein structure prediction including comparative modeling and ab initio protein structure prediction; molecular simulations (molecular dynamics and Monte Carlo simulations).—II, III. Koehl

230. Applied Numerical Linear Algebra (4) Laboratory/discussion—3 hours; discussion—1 hour. Prerequisite: course 130 or Engineering Applied Science 209 or Mathematics 167. Numerical linear algebra [NLA] with emphasis on applications in engineering systems: matrix factorizations; perturbation and rounding error analyses of fundamental NLA algorithms. Offered in alternate years.—I. Bai

231. Large-Scale Scientific Computation (4) Lecture—3 hours; discussion—1 hour. Prerequisite: course 130. Algorithms and techniques for large-scale scientific computation, including basics for high performance computing, iterative methods, discrete and continuous Fourier transform, Poisson solvers, particle methods, spectral graph partition and its applications. Offered in alternate years.—II. Bai

234. Computational Function Genomics (4) Lecture—3 hours; discussion—1 hour. Prerequisite: course 124; graduate standing in Computer Science or Life Sciences. Bioinformatics methods for analysis and inference of functional relationships among genes using large-scale genomic data, including methods for integration of gene expression, promoter sequence, IT-DNA binding and other data, and applications to modeling of biological networks.—II. (II.) Filkov

235A. Computer and Information Security (4) Lecture—3 hours; project. Prerequisite: course 150; course 152A recommended. Modern topics in computer security, including: protection, access control, operating systems security, network security, applied cryptography, cryptographic protocols, secure programming practices, safe languages, mobile code, malware, social engineering, and case studies from real-world systems. Not open for credit to students who have taken course 235.—I. (I.) Chen

235B. Foundations of Computer and Information Security (4) Lecture—3 hours; discussion—1 hour. Prerequisite: course 235A; courses 120, 150 recommended. Theoretical foundations of methods used to protect data in computer and communication systems. Access control matrices and utility of security; policies; Bell-LaPadula, Biba, Chinese Wall models; non-interference and non-deducibility; information flow and the confinement problem. Not open for credit to students who have taken course 235.—II. (II.) Bishop

236. Computer Security: Intrusion Detection Based Approach (4) Lecture—3 hours; discussion—1 hour. Prerequisite: course 150; course 153 recommended. Concepts of intrusion detection, anomaly detection based on machine learning, signature-based detection using pattern matching, automated response to attacks using artificial intelligence planning, tracing intruders based on principal component analysis, security policy languages. Offered in alternate years.—I. Levit

240. Programming Languages (4) Lecture—3 hours; discussion—1 hour. Prerequisites: courses 140A, 142. Advanced topics in programming languages, including formal syntax and semantics, the relation between formal semantics and verification, an introduction to the lambda calculus. Additional topics may include design principles, alternative programming languages, in-depth semantic theory and models of language implementation.—II. (II.) Pandey

242. Translation of Programming Languages (4) Lecture—3 hours; laboratory—3 hours. Prerequisite: course 240. Lexical analysis, parsing, storage management, symbol table design, semantic analysis and code generation. U. LALR grammars. Compiler-compilers.—III. (III.) Pandey

243. Code Generation and Optimization (4) Lecture—3 hours; discussion—1 hour. Prerequisite: course 201A or Engineering Electrical and Computer 270. Compiler optimizations for performance, code size and power reduction. Topics include control- and data-flow analysis, redundancy elimination, loop and cache optimizations, register allocation, local and global instruction scheduling, and modulo scheduling.—II. (II.) Wilken

244. Principles of Concurrent Programming (4) Lecture—3 hours; laboratory—3 hours. Prerequisite: courses 20, 150. Fundamental concepts and applications of concurrent programs; concurrent program verification and derivation; synchronization mechanisms in programming languages; distributed programming techniques in case studies of languages.—I. (I.) Olsson, Pandey, Su

247. Concurrent Programming Languages (4) Lecture—3 hours; laboratory—3 hours. Prerequisite: course 140A, 150. Fundamental concepts and applications of concurrent programs; concurrency control, recovery, security, OS kernels and programming languages, software engineering and operating systems. Models of parallel machines. Load balancing. Scalability. Portability. Efficiency measures. Design and implementation techniques for several classes of concurrent programming languages (such as object-oriented, functional, logic, and constraint-based programming languages).—II. (II.) Pandey, Olsson, Su

251. Operating Systems (4) Lecture—3 hours; discussion—1 hour. Prerequisite: courses 240A, 252A. Design and implementation of operating systems, performance evaluation in operating systems. Algorithms, internal architectures for single processor OS and distributed systems. Concurrency control, recovery, security. OS kernels and programming. Special top- ical embedded systems, real-time systems, device driver, NPU (Network Processor Unit).—III. (III.) Pandey, Wu

252. Computer Networks (4) Lecture—3 hours; laboratory—3 hours. Prerequisite: course 152A. Internet protocol based computer networks, transmission, routing, multime- dia applications and requirements and design.—II. Mukherjee, Mahapatra, Ghosal

255. Resource Management in Wireless Communication Networks (4) Lecture—3 hours; discussion—1 hour. Prerequisite: course 252A. Advanced research issues in wireless communication networks, including multi-user diver- sity and cross-layer optimization, basic network information theory, WMAC protocols and the impact on networks, and dynamics spectrum management. Offered in alternate years.—III. Liu

256. Performance Evaluation (4) Lecture—3 hours; discussion—1 hour. Prerequisite: courses 20, 125A, 154A-154B or Electrical and Computer Engineering 170, Statistics 131A; course 150 recommended. Use of simulation and queueing theory in computer and communication system design. Applications to protocols, channel error rates based on principal component analysis, security policy languages. Offered in alternate years.—I, III. Mat- loff, Ghosal, Mahapatra, Mukherjee

257. Mobile and Wireless Networks (4) Lecture—3 hours, independent study. Prerequisite: course 252. Fundamental techniques in design of second generation wireless networks: cellular net- work and protocols, medium access techniques, handoff control, signaling and mobility manage- ment, wireless data works, Internet mobility and Per- sonal Communication Services (PCS). Third generation wideband systems, novel technologies, adhoc networks. Offered in alternate years.—I. Gha- sal, Mahapatra, Mukherjee


260. Software Engineering (4) Lecture—3 hours, project. Prerequisite: course 142; course 160 recommended. Computer programming techniques for domain-specific software reuse.—I. (I.) Devanbu

261. Program Verification (4) Lecture—3 hours; discussion—1 hour. Prerequisite: Mathematics 125 or Philosophy 112 or familiarity with first-order logic and knowledge of an iterative and functional programming language. Methods of proving correctness of programs with respect to formal specifications, with attention to those suited for employing automated deductive logic (ground, symbolic execution, techniques suited to iterative programming, methods from denotational semantics, termination, dynamic logic and proofs of concurrent programs).—I, III. (I, III.) Levit

Abstract data types, together with methods for speci-
ifying them. Implementations and proofs of implemen-
tation. Use of reason about data and programs. Parameterized types. Constructing good formal specifications. Offered in alternate years.—II.

262. Distributed Database Systems (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 165A. Concepts of distributed database sys-
tems and architectures, distributed database design, distributed query processing and optimization, trans-
actions, and recoveries, concurrent control, heter-
geoneous and multidatabase systems.—I, III. (III.)
Gertz, Ludaescher

266. Spatial Databases (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 152B or 165A. Wide-area distributed infor-
mation systems, data broadcasting, multicast, publish/subscribe, service differentiation, information retrieval, Web caching. Offered in alternate years.—III. Askoy

267. Wide-Area Distributed Information Systems (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 152B or 165A. The challenges confronting the field of vol-
ume visualization. Offered in alternate years.—II. (I.)

268. Scientific Data And Workflow Management (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 165A. Scientific data integration, metadata, knowledge representation, ontologies, scientific work design, and management. Offered in alternate years.—II. (II.)

270. Artificial Intelligence (3)
Lecture—3 hours. Prerequisite: courses 140A, 172. Concepts and techniques underlying the design and implementa-
tion of models of human performance on intelligent tasks and applications of such models. Offered in alternate years.—II.

271. Machine Learning and Discovery (4)
Lecture—3 hours; project—1 hour. Prerequisite: course 170. Artificial intelligence techniques for knowledge acquisition by computers. Fundamental prob-
lems in machine learning and discovery. Systems that learn from examples, analogies, and solved problems. Systems that discover numerical laws and qualitative relationships. Projects centering on implementation and evaluation.—II. Levitt, Vemuri

272. Information Visualization (4)
Lecture—3 hours; laboratory—3 hours. Prerequisite: course 163 or 175 recommended. Advanced topics in in-
formation visualization: perceptually effective display methods, color design and selection, interaction tech-
niques, distortion methods, large graph visualization techniques, visual data mining methods, and evalua-
tion methods.—II. (II.) Ma

273. Applied Visual Computing (4)
Lecture—3 hours; laboratory—3 hours. Prerequisite: graduate standing. Visual computing paradigms, current visualization technologies, principles of 3D graphics, user interface designs, and exploratory visualization. Offered in alternate years.—I, II, III.

274. Automated Deduction (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: Mathematics 125 or Philosophy 112 or familiarity with first-order logic. Techniques of mechanical theo-
rem proving. Methods based on resolution and term-
rewriting. Decision procedures. Induction. Applica-
tions to program verification, question/answering and plan generation. Study existing mechanical the-
orem provers. Offered in alternate years.—III. Levitt

275A. Advanced Computer Graphics (4)
Lecture—3 hours; laboratory—3 hours. Prerequisite: courses 175 or 177 or 178. Advanced topics in com-
puter graphics. Hidden surface models, rendering of various surface types, subdivision methods, shading techniques, anti-aliasing, modeling techniques.—II. (II.)

275B. Advanced Computer Graphics (4)
Lecture—3 hours; laboratory—3 hours. Prerequisite: course 175 or 177 or 178. Advanced topics in com-
puter graphics and geometric modeling. Topics taken from advanced research papers in computer graphics, image analysis, visualization and geometric modeling. Discussion of current research in the field. Offered in alternate years.—II. (II.) Hamann, Ma

276. Advanced Volume Visualization (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 177. Applications, available tools and tech-
niques, the challenges confronting the field of vol-
ume visualization. Considerations of advanced topic in the field. Offered in alternate years.—II. (II.) Hamann, Ma

277. Computer-Aided Geometric Design (4)
Lecture—3 hours; laboratory—3 hours. Prerequisite: course 175. Advanced techniques for the defini-
tion and manipulation of curves and surfaces. Bezier curves and surfaces, B-spline curves and surfaces, subdivision surfaces, wavelets. Integration into vari-
ous computer-aided design systems. Offered in alternate years.—III. (II.) Hamann, Joy, Max

279. Computer Animation (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 175 or 177. Course surveys recent research and fundamental techniques that lie behind charac-
ter animation tools. Emphasis on improving expres-
sive aspects of movement and how physics, motion capture data, the arts and psychology, literature, and

280. Virtual Reality Technology (4)
Lecture—3 hours; discussion—1 hour. Prerequisite: course 175 or 177. Fundamentals and principles of Virtual Reality (VR) technology. Potential and limits for its useful application. Developing a complete virtual reality application. Offered in alternate years.—II.

289. Special Topics in Computer Science (1-5)
Lecture, laboratory, or combination. Prerequisite: consent of instructor. Special topics in (A) Computer Science Theory; (B) Architecture; (C) Programming Languages and Compilers; (D) Operating Systems; (E) Software Engineering; (F) Data Bases; (G) Artifi-
cial Intelligence; (H) Computer Graphics; (I) Net-
works; (J) Computer-Aided Design; (K) Scientific Computing; (L) Computer Science; (M) Network Security; (N) Bioinformatics and Computational Biology. May be repeated for credit when topic changes.—I, II, III. (II.)

290. Seminar in Computer Science (1)
Seminar—1 hour. Participation seminar; discussion and presentation of current research and develop-
manship in computer science. (S/U grading only.)—I, II, III. (II.)

290C. Graduate Research Group Conference (1)
Discussion—1 hour. Research programs, progress and techniques in computer science. May be repeated for credit. (S/U grading only.)—I, II, III. (II.)

293A. Research in Computer Science (1)
Lecture —1 hour. Prerequisite: graduate standing in computer science. Study of research topics in com-
puter science. Ph.D. level research methodologies (experimental, applied and theoretical). Study skills ne-
necessary to successfully find/solve significant research problems. Findings and successful interact-
ing with a research advisor. Ethical issues in research/collaborative work. (S/U grading only).—I.

293B. Research in Computer Science (1)
Lecture—1 hour. Prerequisite: graduate standing in computer science; course 293A recommended. Study of Ph.D. level research methodologies (ex-
perimental, applied and theoretical). Presenting research results for the computer science community.
Study skills necessary to successfully find/solve signifi-
cant research problems. (S/U grading only).—II.

298. Group Study (1-5)
Lecture, laboratory, or combination. Prerequisite: consent of instructor. (S/U grading only.)

299. Research (1-12)
(S/U grading only)

Professional

30. The Teaching of Computer Science (1)
Discussion—1 hour. Prerequisite: meet qualifications for teaching assistant and/or associate-in in Com-
puter Science. Participation as a teaching assistant or associate-in in a designated computer science course. Method of leading discussion groups or laboratory sections, writing and grading quizzes, use of labora-
tory equipment, and grading laboratory reports. May be repeated for credit. (S/U grading only.)—I, II, III. (II.)

306. Teaching Assistant Training Practicum (1-4)
Prerequisite: graduate standing. May be repeated for credit. (S/U grading only.)