3. Professional Development Requirement, Colloquium (3 units). Satisfactory completion of three quarters of CS 287 (Colloquium in Computer Science) or EE 295 (Colloquium in Electrical Engineering) in three distinct quarters.

4. Capstone Experience All students must complete a capstone experience that synthesizes and integrates the knowledge and skills obtained throughout the master’s program, by either passing a comprehensive exam, writing a thesis, or completing a project. The Comprehensive Examination plan is the default option. If a student chooses the alternative project or thesis plan, it is their responsibility to find a faculty member willing to supervise the master’s project or thesis, to form the faculty examining committee, and to schedule the oral examination.

a. Comprehensive Examination (Plan I). A minimum of 36 units are required, of which 24 must be selected from the Core Requirement and Technical Electives courses above. The remaining 12 units must be in approved graduate-level courses related to the major subject area, and/or approved Computer Engineering undergraduate technical electives. Units obtained in graduate research for the thesis or dissertation, directed research, or directed studies (CS 290, CS 297, CS 299, EE 290, EE 297, EE 299) may not be used to satisfy any course requirements under this plan. Students must pass a comprehensive examination administered by the Computer Engineering Program.

b. Project (Plan II). A minimum of 36 units are required, of which 24 must be selected from the Core Requirement and Technical Electives courses above. The remaining 12 units must be in approved graduate-level courses related to the major subject area, and/or approved Computer Engineering undergraduate technical electives, and may include up to 4 units of directed research (CS 297, EE 297) and/or directed studies (CS 290, EE 290). Units obtained in graduate research for the thesis or dissertation (CS 299, EE 299) may not be used to satisfy any course requirements under this plan. Students must complete a research project under the guidance of a faculty member. The project will be approved by a committee of at least two faculty members and requires a presentation and written report.

c. Thesis (Plan III). A minimum of 36 units are required, of which 24 must be selected from the Core Requirement and Technical Electives courses above. The remaining 12 units must be in approved graduate-level courses related to the major subject area, approved Computer Engineering undergraduate technical electives, or graduate research for the thesis or dissertation (CS 299, EE 299). Units obtained in directed research or directed studies (CS 290, CS 297, EE 290, EE 297) may not be used to satisfy any course requirements under this plan. Students must submit a master’s thesis in accordance with the general requirements of the university. The thesis is original research work, and it should demonstrate the student’s ability to study a research area, identify an open problem and make a research contribution. The thesis requires a presentation and must be approved by a committee of at least three faculty members.

Combined B.S. + M.S. Five-Year Program. The college offers a combined five year B.S.+ M.S. program, designed to allow successful UCR Computer Engineering B.S. graduates to complete the Master of Science degree in Computer Engineering in one year, by allowing up to 12 credits of coursework taken as a UCR undergraduate to be counted towards the elective requirements of the M.S. (The courses that can be double counted are those that are eligible to be counted as technical electives in the B.S. requirements.)

A student may apply at the start of their senior year by submitting an application to the Computer Engineering M.S. program. Provided that at the end of junior year, the student was a UCR Computer Engineering B.S. student with cumulative GPA at least 3.4 and had completed the following courses with no grade less than a B- and average grade at least 3.2: CS 100, CS 120A, CS 120B, CS 161, CS 161L. The application to the M.S. program must include at least two recommendation letters from UCR Academic Senate faculty members (at least one, and preferably both, CSE faculty). Submission of GRE scores with the application is recommended but not required. Matriculation into the combined program occurs in the Fall term following senior year, provided: (a) the M.S. application is accepted, (b) throughout senior year, the student is a Computer Engineering B.S. major with cumulative GPA 3.4 or higher, (c) by the end of senior year, the student completes the Computer Engineering B.S. degree requirements.

Incoming students who are applying to the Computer Engineering B.S. program may simultaneously apply for preliminary admission into the combined program provided their high school GPA is at least 3.6, their SAT-I combined score is at least 1530, they satisfy the Entry Level Writing requirement before matriculation, and they have sufficient math preparation to enroll in calculus upon arrival. Preliminary admission status is maintained as long as the student is a Computer Engineering or Computer Science B.S. student in good standing with a cumulative GPA of at least 3.4. Preliminary admitted students still need to apply for full admission in their senior year as described above.

Five-year programs leading to M.S. degrees in other programs (including Computer Science) are also available. They are described separately in the catalog sections for those programs.

4. Computer Science and Engineering

Subject abbreviation: CS
The Marian and Rosemary Bourns College of Engineering

Marek Chrobak, Ph.D., Chair

Department Office, 351 Winston Chung Hall
(951) 827-5639; cs.ucr.edu

Distinguished Professor
Laxmi N. Bhuyan, Ph.D.

Professors
Marek Chrobak, Ph.D.
Nael Abu-Ghazaleh, Ph.D.
Rajiv Gupta, Ph.D.
Tao Jiang, Ph.D.
President’s Chair
Eamonn Keogh, Ph.D.
Srikanth Krishnamurthy, Ph.D.
Stefano Lonardi, Ph.D.
Marin Mole, Ph.D.
Walid Najjar, Ph.D.
Michael Pazzani, Ph.D.
Kadangoole K. Ramakrishnan, Ph.D.
Chinya Ravishankar, Ph.D.
Vassilis Tsotras, Ph.D.
Frank N. Vahid, Ph.D.
Neal Young, Ph.D.

Professors Emeriti
Yang-Chang Hong, Ph.D.
Thomas H. Payne, Ph.D.
Teodor C. Przymundski, Ph.D.

Associate Professors
Evangelos Christidis, Ph.D.
Iulian Neamtu, Ph.D.
Christian Shelton, Ph.D.
Victor Zordan, Ph.D.

Assistant Professors
Philip Brisk, Ph.D.
Zichong Chen, Ph.D.
Zhiyuan Qian, Ph.D.
Tamar Shinar, Ph.D.

Adjunct Professors
Gianfranco Ciardo, Ph.D.
Michalis Faloutsos, Ph.D.

Cooperating Faculty
Bir Bhanu, Ph.D.
(I Electrical and Computer Engineering)
Ilya Dumir, Ph.D.
(I Electrical and Computer Engineering)
Thomas Girke, Ph.D.
(Plant and Plant Sciences)
Roger Lake, Ph.D.
(I Electrical and Computer Engineering)
Michel L. Lapudis, Ph.D.
(Mathematics)
Erik Rolland, Ph.D.
(Accounting and Information Systems)
Amir Roy-Chowdhury, Ph.D.
(I Electrical and Computer Engineering)
Thomas Stahovich, Ph.D.
(Mechanical Engineering)
Sheldon Tan, Ph.D.
(I Electrical and Computer Engineering)
Qi Zhu, Ph.D.
(I Electrical and Computer Engineering)

Major
The Department of Computer Science and Engineering offers three majors at the undergraduate level. UCR’s offerings of all three majors are unique compared to many schools in the emphasis on theoretical foundations and practical applications.

The Computer Science major stresses the study of core and advanced computer science topics. It prepares students for a large variety of careers in computing, including software
All undergraduates in the College of Engineering must see an advisor at least annually. Visit student.engr.ucr.edu for details.

University Requirements
See Undergraduate Studies section.

College Requirements
See The Marlan and Rosemary Bourns College of Engineering, Colleges and Programs section.

The Computer Science major uses the following major requirements toward the satisfaction of some of the college’s Natural Sciences and Mathematics breadth requirements and one of the college’s English Composition breadth requirements.

1. **ENGR 180W**
2. **MATH 008B or MATH 009A**
3. **PHYS 040A, PHYS 040B, PHYS 040C**

The Business Informatics major uses the following major requirements toward the satisfaction of the college’s Social Sciences breadth requirements and one of the College’s Natural Science and Mathematics breadth requirements.

1. **ECON 002, ECON 003**
2. **MATH 008B or MATH 009A**
3. **SOC 150**

Major Requirements

**Computer Science Major**

1. **Lower-division requirements (61 units)**
   a) ENGR 001-I
   b) CS 010 or CS 010V, CS 012 or CS 012V or CS 013, CS 014, CS 061
   c) CS 011/MATH 011
   d) MATH 008B or MATH 009A, MATH 009B, MATH 009C, MATH 010A, MATH 031
   e) PHYS 040A, PHYS 040B, PHYS 040C
   f) One course of 4 or more units in an engineering discipline outside the field of computer science to be selected in consultation with a faculty advisor. (Either a lower-division or an upper-division course may be used to satisfy this requirement.)

2. **Upper-division requirements (86 units minimum)**
   a) ENGR 101-I
   b) CS 100, CS 141, CS 150, CS 152, CS 153, CS 161, CS 179 (E-Z)
   c) CS 120A/ECE 120A, CS 120B/ECE 120B
   d) CS 111/MATH 111
   e) ENGR 180W
   f) STAT 155
   g) Two courses from MATH 046, MATH 120, MATH 126, PHI 124
   h) At least 28 units of technical electives to be chosen from an approved list of courses which currently includes CS 122A, CS 122B, CS 130, CS 133, CS 134, CS 145, CS 151, CS 160, CS 162, CS 164, CS 165, CS 166, CS 168, CS 169, CS 170, CS 171, CS 172, CS 177, CS 179 (E-Z) (4 units maximum), EE 140, MATH 120, MATH 135A, MATH 135B. The technical electives selected must be distinct from those used to satisfy the requirements specified in 2.a)–h) above.

Business Informatics Major

1. **Lower-division requirements (56 units)**
   a) ENGR 001M
   b) BUS 020
   c) CS 010 or CS 010V, CS 012 or CS 012V, or CS 013, CS 014, CS 061
   d) CS 011/MATH 011
   e) ECON 002, ECON 003
   f) MATH 008B or MATH 009A, MATH 009B, MATH 009C, MATH 010A, and MATH 031

2. **Upper-division requirements (93 units)**
   a) ENGR 101M
   b) BUS 103, BUS 104/STAT 104, BUS 106/ECON 134
   c) CS 100, CS 141, CS 153, CS 165
   d) At least two courses from CS 164, CS 166, CS 172, CS 180
   e) CS 111/MATH 111
   f) ENGR 180W
   g) SOC 150
   h) STAT 155
   i) Sixteen (16) units of upper-division Computer Science technical electives, which must be distinct from the courses used to satisfy the above major requirements. These 16 units may be chosen from those courses listed as upper-division requirements or technical electives for the Computer Science major. At least three courses must be in the Department of Computer Science and Engineering.
   j) Twenty-four (24) units of Business Administration technical electives, including at least 8 units of courses listed in the Information Systems concentration within the Business Administration major. These 24 units must be distinct from the courses used to satisfy the above major requirements and may be chosen from any of the available Business Administration courses, with the following restrictions: no credit will be given for BUS 101, only one of BUS 171 and CS 180 can be taken for credit, only one of BUS 173 and CS 166 can be taken for credit, only one of BUS 175 and CS 164...
can be taken for credit, and only one of BUS 125 and CS 177 can be taken for credit. Students may petition for exceptions to the above degree requirements.Exceptions to Computer Science course requirements must be approved by the Computer Science and Engineering undergraduate advisor or chair. Visit the Student Affairs Office in the College of Engineering or student.engr.ucr.edu for a sample program.

Minor in Computer Science
The minor in Computer Science is designed to enhance majors with limited computational theory or practice. As such, students with majors in Computer Engineering, Computer Science, Business Informatics, and Mathematics (Computational Mathematics option) are not eligible.

Requirements for the minor in Computer Science are:
1. Lower-division courses: CS 010 or CS 010V, CS 012 or CS 012V, or CS 013, CS 014, CS 061, CS 011/MATH 011, MATH 008B or MATH 009A, MATH 009B, MATH 009C
2. Core courses: CS 100, CS 111
3. Three elective courses, each of four or more units, such that:
   a) Each is an upper-division requirement or a listed technical elective for the Computer Science major, excluding courses numbered 190-199
   b) No course may be an upper-division requirement of the student’s major
   c) At least two courses must be in the Department of Computer Science and Engineering
4. All courses for the minor must be taken for a letter grade.

Note Students with a minor in Computer Science must obtain approval from the undergraduate advisor in Computer Science and Engineering for a specific program of electives consistent with their career goals.

Graduate Program
The Department of Computer Science and Engineering offers the M.S. and Ph.D. degrees in Computer Science. General requirements are listed in the Graduate Studies section of this catalog. Specific requirements for each degree are described below. Students enrolled prior to Fall 2008 can still follow the old Graduate Program.

Admission All applicants must supply GRE General Test scores. The GRE subject test in Computer Science is recommended but not required. Applicants should have at least an undergraduate degree in computer science or a closely related field, but applicants who fail to meet this criterion may sometimes be admitted with deficiencies.

Prerequisite Material Competence in the areas defined by the following UCR courses is essential to graduate study in computer science: CS 141, CS 150, CS 152, CS 153, CS 161. A student who is deficient in any of these competency areas may be asked to complete the corresponding UCR course with a letter grade of at least B+, or to pass a challenge examination based on that course’s final exam with a grade of at least B+. All such remedial work should be completed within the first year of graduate study, and in all cases the deficiency must be corrected before a student can enroll in any graduate course from the same specialty area.

Core Areas Students have considerable flexibility in selecting specialty area(s) within the program. However, the following core areas introduce fundamental concepts and tools of general interest to all students.
1. Hardware design principles: CS 203 or CS 220.
2. Theoretical foundations: CS 215 or CS 218.

Major Specialty Areas The department has active research programs in the following major specialty areas. A list of related graduate courses is provided for each area. Courses that qualify for the M.S. Breadth Requirement are marked with an asterisk (*).
A. Algorithms, Bioinformatics, and Theory of Computation: CS 219*, CS 218*, CS 234, CS 238
C. Databases, Data Mining, and Machine Learning: CS 205*, CS 235*, CS 229, CS 236*, CS 272
E. Computer Networks: CS 204*, CS 237, CS 239*, CS 240, CS 257, CS 255*
F. Programming Languages, Compilers, and Software Engineering: CS 201*, CS 206*, CS 207*, CS 245*, CS 246*

Master’s Degree
The Department of Computer Science and Engineering offers the M.S. degree in Computer Science, after completion of the following degree requirements.

Satisfactory completion of CS 287 (Colloquium in Computer Science) each quarter of enrollment for full-time in-residence graduate students.

Course Requirements 48 quarter units of graduate or upper-division undergraduate courses are required. Students who have completed similar courses elsewhere may petition for a waiver of a required course or for substitution of an alternative course. For students interested in interdisciplinary research, individual study programs can be approved. All courses used to satisfy these requirements (with the exception of CS 297 and CS 299) must be taken for a letter grade. No course can be counted towards more than one category.

1. Core Requirement (8 units) Choose one course from two of the three Core Areas listed above, with no grade lower than B-.

2. Breadth Requirement (8 units) Two approved breadth courses chosen in such a way that together the core and breadth courses cover four different Major Specialty Areas (A to G).

3. Electives (32 units) Students have the option of completing their degree by taking a comprehensive exam, writing a thesis, or completing a project. Depending on the option selected, the electives that may be taken are:
   a. Comprehensive Examination Option. For a student pursuing the M.S. degree, comprehensive examination option, the 32 elective units must include at least 16 units of approved graduate lecture courses. The remaining 16 units may include additional approved graduate lecture courses, up to 8 units of graduate seminars in CS 260–269, and up to 12 units of approved undergraduate technical electives. Research units (CS 297 or CS 299) may not be used to satisfy any course requirements under this option.
   b. Project Option. A student pursuing the M.S. degree, project option, may include up to 4 units of Directed Research (CS 297) towards the elective requirement. Of the remaining 28 units, at least 12 units must be approved graduate lecture courses. The remaining 16 units may include additional approved graduate lecture courses, up to 8 units of graduate seminars in CS 260–269, and up to 12 units of approved undergraduate technical electives.
   c. Thesis Option. A student pursuing the M.S. degree, thesis option, may include up to 12 units of graduate research (CS 297 or CS 299) towards the elective unit requirement. Of the remaining 20 units, at least 4 units must be approved graduate lecture courses. The remaining 16 units may include additional approved graduate lecture courses, up to 8 units of graduate seminars in CS 260–269, and up to 8 units of approved undergraduate technical electives.

Capstone Experience All students must complete a capstone experience that synthesizes and integrates the knowledge and skills obtained throughout the master’s program, by either passing a comprehensive exam, writing a thesis, or completing a project. The Comprehensive Examination Option is the default option. If a student chooses the project or thesis option, it is the responsibility of the student to find a faculty member willing to supervise the master’s project or thesis, to form the faculty examining committee, and to schedule the oral examination.

a. Comprehensive Examination Option Students must pass a comprehensive examination administered by the
Department of Computer Science and Engineering.

b. Project Option Students must complete a research project under the guidance of a faculty member. The project will be approved by a committee of at least two faculty members and requires a presentation and written report.

c. Thesis Option Students must submit a master’s thesis in accordance with the general requirements of the university. The thesis is original research work, and it should demonstrate the student’s ability to study a research area, identify an open problem, and make a research contribution. The thesis requires a presentation and must be approved by a committee of at least three faculty members.

Normative Time to Degree 2 years.

Combined B.S. + M.S. Five-Year Program The college offers combined five-year B.S. + M.S. programs designed to allow successful UCR Computer Science or Computer Engineering B.S. graduates to complete the Master of Science degree in Computer Science in one year, by allowing up to 12 credits of coursework taken as a UCR undergraduate to be counted towards the 32-unit elective requirements of the M.S. (The courses that can be double counted are those that are eligible to be counted as technical electives in the B.S. requirements.)

A student may apply at the start of their senior year by submitting an application to the Computer Science M.S. program, provided that at the end of junior year, the student was a UCR Computer Science or Computer Engineering B.S. student with cumulative GPA at least 3.4 and had completed the following courses with no grade less than a B- and average grade at least 3.2: CS 100, CS 120A, CS 120B, CS 161. The application to the M.S. program must include at least two recommendation letters from UCR Academic Senate faculty members (at least one, and preferably both, CSE faculty). Submission of GRE scores with the application is recommended but not required. Matriculation into the combined program occurs in the Fall term following senior year, provided: (a) the M.S. application is accepted, (b) throughout senior year, the student is a Computer Science or Computer Engineering B.S. major with cumulative GPA 3.4 or higher, (c) by the end of senior year, the student completes the Computer Science or Computer Engineering B.S. degree requirements.

Incoming students who are applying to the Computer Science or Computer Engineering B.S. programs may simultaneously apply for preliminary admission into the combined program provided their high school GPA is at least 3.6, their SAT-I combined score is at least 1950, they satisfy the Entry Level Writing requirement before matriculation, and they have sufficient math preparation to enroll in calculus upon arrival. Preliminary admission status is maintained as long as the student is a Computer Science or Computer Engineering B.S. student in good standing with a cumulative GPA of at least 3.4. Preliminary admitted students still need to apply for full admission in their senior year as described above.

Five-year programs leading to M.S. degrees in other programs (including Computer Engineering) are also available. They are described separately in the catalog sections for those programs.

Doctoral Degree

The Department of Computer Science and Engineering offers the Ph.D. degree in Computer Science, after completion of the following degree requirements. It provides a research-oriented education in preparation for a career in research, industry, or academia and exploring both the fundamental aspects of computer science and engineering as well as their applications.

Satisfactory completion of CS 287 (Colloquium in Computer Science) each quarter of enrollment for full-time in-residence graduate students.

Course Work The course requirements for the Ph.D. degree ensure that Ph.D. students are exposed to fundamental concepts and tools (core requirement), a deep up-to-date view of their research specialty area (depth requirement), and an advanced, up-to-date view of the same topics outside their area (breadth requirement). Students are expected to complete all of these course requirements in the first two years of the program. These requirements consist of 44 quarter units of approved graduate or upper-division undergraduate courses, satisfying all four of the following course work categories. All of these courses must be taken for a letter grade, and no course can be counted towards more than one category. Students who have completed similar courses elsewhere may petition for a waiver of a required course or for substitution of an alternative course.

Units obtained in CS 270, CS 287, CS 290, CS 297, CS 298, CS 299, CS 301, and CS 302 cannot be counted in any course work category.

1. Core Requirement (12 units). Choose three courses from at least two of the three Core Areas described above, with no grade lower than a B- and an overall core course GPA of at least 3.2.

2. Depth Requirement (8 units). Choose two courses listed above under the same Major Area (A to G). This requirement ensures that Ph.D. students, early on in their careers, acquire some depth of knowledge in a particular research area.

3. Breadth Requirement (12 units). Choose three courses from at least two different Major Areas (A to G) outside the student’s depth area. No course that is listed in the student’s depth area can be used to fulfill the breadth requirement, even if it is cross-listed in another area. Students, with the consent of the major professor, may petition for a non-CSE course to be counted towards the breadth requirement.

4. Electives (12 units). The remaining courses can be selected from additional

CS graduate lecture courses, up to 8 units of graduate seminars in CS 260-269, and up to 8 units of approved undergraduate technical electives. Students, with the consent of the major professor, may petition for a non-CSE course to be counted as an elective.

Milestones

The Department has established three milestones to mark progress towards the Ph.D. degree in Computer Science: advancement to candidacy, presentation of the dissertation proposal, and final oral examination. A Ph.D. student must also satisfy all applicable Graduate Division requirements for each milestone.

Milestone I: Advancement to Candidacy. A student advances to candidacy after he/she has completed all of the Ph.D. course requirements described above, and passed the combined written and oral qualifying examinations, as described below. These two exams are 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 dissertation; (2) ability to perform critical study, analysis and writing in a focused area; and (3) demonstrated research experience or ability to do research.

Written Qualifying Examination The written qualifying examination consists of a written report summarizing the oral presentation to be given at the oral qualifying examination. This report must be written in proper technical English and in the style of a typical Computer Science conference or journal publication, and must be submitted to the Qualifying Committee for approval at least one week prior to the oral qualifying examination.

Oral Qualifying Examination The student is expected to demonstrate research aptitude by undertaking a research study on some topic (typically a problem from student’s chosen research specialty that may be a promising area in which to conduct the dissertation research), under the guidance of his or her faculty major professor. The research must be presented orally to a Qualifying Committee, which is appointed by the Graduate Division based on nominations from the department. The committee will consist of at least four Senate faculty members, with at least three members whose home department is CSE. The committee evaluates the merits of the work and the student’s aptitude for research. The work must represent significant progress towards original and publishable research. The student must complete this requirement in no more than two attempts. The normative time for taking the Oral Qualifying Exam is by the end of the second year.

Dissertation Committee After advancing to candidacy, the student must form a Doctoral Examination Committee chaired by his or her major professor. The committee will consist of at least four Senate faculty members with at least three members belonging to the CSE department (the other department is CSE).

Milestone II: Dissertation Proposal Examination After advancement to candidacy, the student prepares a dissertation proposal that describes the dissertation topic, summarizes the
CS 010V Introduction to Computer Science for Science, Mathematics, and Engineering (4) Lecture, 1 hour; discussion, 2 hours; written work, 6 hours. Prerequisite(s): a college mathematics course (may be taken concurrently) or credit for Math 009A from the Advanced Placement Examination or the Mathematics Advisory Examination. Covers problem solving through structured programming of algorithms on computers using the C++ object-oriented language. Includes variables, expressions, input/output (I/O), branches, loops, functions, parameters, arrays, strings, file I/O, and classes. Also covers software design, testing, and debugging. An online instruction approach through a weekly webcast lecture and online forum discussion sessions. Credit is not awarded for CS 010V if it has already been awarded for CS 010 or CS 030.

CS 011 Introduction to Discrete Structures (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): MATH 009A (or MATH 09HA); CS 010 or CS 010V or MATH 009B (or MATH 09HB). Introduction to basic concepts of discrete mathematics emphasizing applications to computer science. Topics include prepositional and predicate calculus, elementary set theory, functions, relations, proof techniques, elements of number theory, enumeration, and discrete probability. Cross-listed with MATH 011.

CS 012 Introduction to Computer Science for Science, Mathematics, and Engineering II (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 010 or CS 010V with a grade of "C" or better; familiarity with C or C++ language. Covers structured and object-oriented programming in C++. Emphasizes good programming principles and development of substantial programs. Topics include recursion, pointers, linked lists, abstract data types, and libraries. Also covers software engineering principles. Credit is awarded for only one of CS 012 or CS 012V or CS 013.

CS 012V Introduction to Computer Science for Science, Mathematics, and Engineering II (4) Lecture, 1 hour; discussion, 2 hours; written work, 6 hours. Prerequisite(s): CS 010 or CS 010V with a grade of "C" or better; familiarity with the C or C++ language. Covers structured and object-oriented programming in C++. Emphasizes good programming principles and development of substantial programs. Topics include recursion, pointers, linked lists, abstract data types, and libraries. Also covers software engineering principles. Uses an online instruction approach through a weekly webcast lecture and online forum discussion sessions. Credit is awarded for only one of CS 012V or CS 012 or CS 013.

CS 013 Introductory Computer Science for Engineering Majors (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 010 or CS 010V with a grade of "C" or better; familiarity with C or C++ language. Covers structured and object-oriented programming in C++. Emphasizes good programming principles and development of substantial programs. Topics include recursion, pointers, linked lists, abstract data types, and libraries. Covers software engineering principles. Utilizes examples and assignments specific to engineering disciplines such as numerical data analysis, matrix computations, and dynamic systems. Credit is awarded for only one of CS 012V or CS 012 or CS 013.

CS 014 Introduction to Data Structures and Algorithms (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 012 with a grade of "C" or better or CS 013 with a grade of "C" or better, proficiency in C++. Topics include basic data structures such as arrays, lists, stacks, and queues. Covers data structures and algorithms that meet the requirements of the C++ standard library, and development tool-chain use. Graded Satisfactory (S) or No Credit (NC). Each segment is repeatable as topics change to a maximum of 12 units.

CS 015Y Python (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 010 or CS 010V or knowledge of programming or consent of instructor. A practical exploration of problem solving using Python in a laboratory setting. Focuses on syntax, concepts, and development tool-chain use. Graded Satisfactory (S) or No Credit (NC). Segment is repeatable as topics change to a maximum of 12 units.

CS 016 Machine Organization and Assembly Language Programming (4) Lecture, 3 hours; laboratory, 3 hours. Individual study, 2 hours. Covers basic skills necessary to create three-dimensional digital images and models. Emphasizes techniques for polygon and curved-surface modeling and photorealistic image creation through material shading, texturing, and lighting. Introduces basic scripting methods to create complex models and images with Autodesk Maya or equivalent. Normally graded Satisfactory (S) or No Credit (NC), but students may petition the instructor for a letter grade on the basis of assigned extra work or examination. Cross-listed with ART 066.

CS 017 Three-Dimensional Digital Modeling and Animation (4) Lecture, 3 hours; laboratory, 3 hours; individual study, 2 hours. Prerequisite(s): ART 066/CS 066. Builds advanced skills for three-dimensional modeling. Introduces basic computer animation techniques within framework of existing software. Techniques include rigging, keyframing, and special effects animation using Autodesk Maya software or equivalent. Teaches proficiency in analogous scripting operations.
Normally graded Satisfactory (S) or No Credit (NC), but students may petition the instructor for a letter grade on the basis of assigned extra work or examination. Course is repeatable to a maximum of 8 units. Cross-listed with ART 067.

Upper-Division Courses

CS 100 Software Construction (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 014 with a grade of “C-” or better. Emphasizes development of software systems. Topics include design and implementation strategies; selection and mastery of programming languages, environment tools, and development processes. Develops skill in programing, testing, debugging, performance evaluation, component integration, maintenance, and documentation. Covers professional and ethical responsibilities and the need to stay current with technology.

CS 111 Discrete Structures (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 010 or CS 010V, CS 011/MATH 011; MATH 009C (or MATH 09HC). A study of discrete mathematical structures emphasizing applications to computer science. Topics include asymptotic notation, generating functions, recurrence equations, elements of graph theory, trees, algebraic structures, and number theory.

CS 120A Logic Design (5) Lecture, 3 hours; laboratory, 3 hours; individual study, 3 hours. Prerequisite(s): CS 063 with a grade of “C-” or better. Covers design of digital systems. Includes Boolean algebra, combinational and sequential logic design; and use of arithmetic logic units, carry-lookahead adders, multiplexers, decoders, comparators, multipliers, flip-flops, registers, and simple memories; state-machine design; and basic register-transfer level design. Uses hardware description languages, synthesis tools, programmable logic, and significant hardware prototyping. Cross-listed with EE 120A.

CS 120B Introduction to Embedded Systems (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 120A/EE 120A. Introduction to hardware and software design of digital computing systems embedded in electronic devices (e.g., digital cameras or portable video games). Includes embedded processor programming, microcontroller design, standard peripherals, memories, interfacing, and hardware/software tradeoffs. Involves use of synthesis tools, programmable logic, microcontrollers, and developing working embedded systems. Cross-listed with EE 120B. Credit is awarded for only one of CS 121 or CS 120B/EE 120B.

CS 121 Programming Embedded Systems (4) Discussion, 2 hours; written work, 6 hours. Prerequisite(s): CS 010 or CS 010V and CS 061, or consent of instructor. An online study of the programming of embedded computer systems involving C language and microcontrollers. Includes an introduction to embedded systems; overview of C programming; bit-level manipulation; and capturing time-oriented behavior using synchronous state machines. Covers input/output, concurrency, creating a simple task scheduler, task communication, utilization and scheduling, and coding issues. Credit is awarded for only one of CS 121 or CS 120B/EE 120B.

CS 121L Laboratory in Programming Embedded Systems (2) Discussion, 1 hour; laboratory, 3 hours; written work, 6 hours. Prerequisite(s): CS 121 (may be taken concurrently), or consent of instructor. An online hands-on embedded systems programming training using microcontrollers and simple input/output devices. Covers bit-level manipulation, time-oriented programming using timers, input/output, concurrency, creating a simple task scheduler, task communication, utilization and scheduling, and coding issues. Credit is awarded for only one of CS 121L or CS 120B/EE 120B.

CS 122A Intermediate Embedded and Real-Time Systems (5) Lecture, 3 hours; laboratory, 6 hours. Prerequisite(s): CS 012 or CS 012V or CS 013; CS 120B/EE 120B. Covers software and hardware design of embedded computing systems. Includes hardware and software co-design, advanced programming paradigms (including state machines and concurrent processes), real-time programming and operating systems, basic control systems, and modern chip and design technologies. Laboratories involve use of microcontrollers, embedded microprocessors, programmable logic, and embedded systems. Covers real-time simulation, and debugging environments.

CS 122B Advanced Embedded and Real-Time Systems (5) Lecture, 3 hours; laboratory, 6 hours. Prerequisite(s): CS 122A. Explores state-of-the-art aspects of building embedded computer systems. Topics include real-time programming, synthesis of coprocessor cores, application-specific processors, hardware and software co-design, low-power design, reconfigurable computing, core-based design, and platform-based methodology.

CS 130 Computer Graphics (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 100, MATH 031 (MATH 031 may be taken concurrently); or consent of instructor. A study of the fundamentals of computer graphics necessary to design and build graphics applications. Examines raster graphics algorithms including scan-converting graphics primitives, anti-aliasing, and clipping. Also covers geometric transformations, viewing, solid modeling techniques, hidden-surface removal algorithms, color models, illumination, and shading.

CS 133 Computational Geometry (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 100, CS 111, MATH 031, or equivalents. An introduction to the design of geometry algorithms. Covers the basic computational geometry concepts and techniques used in graphics, robotics, and engineering design. Topics include polygons and polytopes, convex hulls, and voronoi diagrams.

CS 134 Video Game Creation and Design (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 130. Covers academic, theoretical, and practical aspects of video games by exploring common algorithms, data structures, and software design for different genres. Topics include game interface, character movement, intelligent behaviors, and networked or multiplayer games. Requires in-depth, applied programming and a term project, including the design, implementation, and analysis of a computer game.

CS 141 Intermediate Data Structures and Algorithms (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 014 or CS 014V. CS 111; MATH 009C or MATH 09HC; proficiency in C++. Explores basic algorithm analysis using asymptotic notations, summation and recurrence relations, and algorithms and data structures for discrete structures including trees, strings, and graphs. Also covers general algorithm design techniques including “divide-and-conquer,” the greedy method, and dynamic programming. Integrates knowledge of data structures, algorithms, and programming.

CS 145 Combinatorial Optimization Algorithms (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 141; MATH 031, 032; or consent of instructor. Study of efficient algorithm design techniques for combinatorial optimization problems. Topics include shortest paths, minimum spanning trees, network flows, maximum matchings, stable matchings, linear programming, duality, two-person games, algorithmic techniques for integer programming problems, NP-completeness, and approximation algorithms.

CS 150 Automata and Formal Languages (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 014 with a grade of “C-” or better; CS 111; MATH 009C or MATH 09HC. A study of formal languages. Includes regular and context-free languages; computational models for generating these languages such as finite-state automata, pushdown automata, regular expressions, and context-free grammars; mathematical properties of the languages and models; and

equivalence between the models. Also introduces Turing machines and decidability.

CS 151 Introduction to Theory of Computation (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 141, CS 150. The study of fundamental questions about the nature of computing. Topics include Turing machines, computability, reductions, complexity theory, completeness classes P, NP, the P=NP problem, NP-completeness, and other time and space complexity classes.

CS 152 Compiler Design (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 061, CS 100, CS 111. Covers the fundamentals of compiler design. Includes lexical analysis, parsing, semantic analysis, compile-time memory organization, run-time memory organization, code generation, and compiler portability issues.

CS 153 Design of Operating Systems (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 061, CS 100, CS 111; C++ programming proficiency. Covers the principles and practice of operating system design. Includes concurrency, memory management, file systems, protection, security, command languages, scheduling, and system performance.

CS 160 Concurrent Programming and Parallel Systems (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 061, CS 100, CS 111. A study of concurrent and parallel systems. Topics include modular structure and design, interprocess communication, synchronization, failures, persistence, and concurrrency control. Also covers atomic transactions, recovery language support, distributed interprocess communication, and implementation mechanisms. Provides preparation for the study of operating systems, databases, and computer networking.

CS 161 Design and Architecture of Computer Systems (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 120A/EE 120A; concurrent enrollment in CS 161L. A study of the fundamentals of computer design. Topics include the performance evaluation of microprocessors; instruction set design and measurements of use; microprocessor implementation techniques including multicycle and pipelined implementations; computer arithmetic; memory hierarchy; and input/output (I/O) systems.

CS 161L Laboratory in Design and Architecture of Computer Systems (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 120A/EE 120A; concurrent enrollment in CS 161L. Covers the design and simulation of a complete computer system using hardware description language and simulator. Topics include instruction set architecture design; assemblers; datapath and control unit design; arithmetic and logic unit; memory and input/output (I/O) systems; and integration of all parts into a working computer system.

CS 162 Computer Architecture (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 161 and CS 161L with grades of “C-” or better. The study of advanced processor design. Topics include CPU pipeline, data and control hazards, instruction-level parallelism, branch prediction, and dynamic scheduling of instructions. Also covers very Long Instruction Word (VLIW) processing, multimedia support, design of network and embedded processors, cache partitioning and design, shared memory and message passing, and network topologies.

CS 164 Computer Networks (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 100, CS 111, CS 153. Covers the fundamentals of computer networks. Topics include layered network architecture, communication protocols, local area networks, UNIX network programming, verification, network security, and performance studies.

CS 165 Computer Security (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 141, CS 153. Examines the ways in which information systems
are vulnerable to security breaches. Topics include attacks; security labels, lattices, and policies; safeguards and countermeasures; intrusion detection; authorization and encryption techniques; networks; digital signatures, certificates, and passwords; privacy issues, firewalls, and spoofing; Trojan horses and computer viruses; CERT Coordination Center; and electronic commerce.

CS 166 Database Management Systems (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 100, CS 111. Covers architecture of database management systems, relational systems, and hierarchical models; distributed database concepts; query language implementation issues; and privacy and security of the database.

CS 168 Introduction to Very Large Scale Integration (VLSI) Design (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 120A/EE 120A or consent of instructor. Studies integrated circuit fabrication, device characterization, and circuit simulation. Introduces basic device physics and physical design rules, MOS logic design, and timing and clock schemes. Covers layout generation, subsystem designs, and circuits for alternating logic styles. Also covers design and simulation using hardware description language and CAD tools. Cross-listed with EE 168.

CS 169 Mobile Wireless Networks (4) Lecture, 3 hours; laboratory, 2 hours; extra reading, 1 hour. Prerequisite(s): CS 153 or consent of instructor. Introduces the fundamentals of wireless and mobile networks. Covers wireless channel models, MAC protocols, and wireless networking. Also covers cellular, WLAN and ad hoc networks, and routing in multi-hop wireless networks. Includes wireless security and the impact of wireless links on TCP and other transport layer solutions.

CS 170 Introduction to Artificial Intelligence (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 100 and CS 111. An introduction to the field of artificial intelligence. Focuses on discrete-value problems. Covers heuristic search, problem representation, and classical planning. Also covers constraint satisfaction and logical inference.

CS 171 Introduction to Machine Learning and Data Mining (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 100, CS 111. Introduces formalisms and methods in machine learning and data mining. Topics include data representation, supervised learning, and classification. Covers regression and clustering. Also covers rule learning, function approximation, and margin-based methods.

CS 172 Introduction to Information Retrieval (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 100, CS 111; EE 144 or STAT 155. Introduces information retrieval (IR) principles and techniques for indexing and searching document collections. Topics include Web search, text processing, ranking algorithms, search in social networks, and search evaluation. Also studies scalability issues in search engines. Satisfactory (S) or No Credit (NC) grading is not available.

CS 177 Modeling and Simulation (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 100, CS 111, C++ programming proficiency. Covers validation of random number sequences; concepts in modeling and systems analysis; and conceptual models and their mathematical and computer realizations. Examines simulation modeling techniques, including object-oriented modeling and discrete-event modeling. Emphasizes the use of simulation libraries used with programming languages such as C++.

CS 179 (E-Z) Project in Computer Science (4) For hours and prerequisites, see segment descriptions. Under the direction of a faculty member, student teams propose, design, build, test, and document software and/or hardware devices or systems. Emphasizes professional and ethical responsibilities and the need to stay current on technology and its global impact on economics, society, and the environment.

CS 178 Compilers (4) Discussion, 1 hour; laboratory, 9 hours. Prerequisite(s): CS 100 and CS 152 with grades of “C-” or better; ENGR 180W; 8 additional upper-division units in Computer Science. Covers the planning, design, implementation, testing, and documentation of a compiler-related system. Incorporates techniques from previous related courses. Emphasizes professional and ethical responsibilities; the need to stay current on technology; and its global impact on economics, society, and the environment.

CS 179F Operating Systems (4) Discussion, 1 hour; laboratory, 9 hours. Prerequisite(s): CS 153 with a grade of “C-” or better; ENGR 180W; 8 additional upper-division units in Computer Science. CS 160 is recommended. Covers the planning, design, implementation, testing, and documentation of an operating systems-related system. Incorporates techniques from previous related courses. Emphasizes professional and ethical responsibilities; the need to stay current on technology; and its global impact on economics, society, and the environment.

CS 179G Database Systems (4) Discussion, 1 hour; laboratory, 9 hours. Prerequisite(s): CS 100 and CS 166 with grades of “C-” or better; ENGR 180W; 8 additional upper-division units in Computer Science. Covers the planning, design, implementation, testing, and documentation of a database-related system. Incorporates techniques from previous related courses. Emphasizes professional and ethical responsibilities; the need to stay current on technology; and its global impact on economics, society, and the environment.

CS 179-I Networks (4) Discussion, 1 hour; laboratory, 9 hours; laboratory, 3 hours. Prerequisite(s): CS 100 and CS 164 with grades of “C-” or better; ENGR 180W; 8 additional upper-division units in Computer Science. Covers the planning, design, implementation, testing, and documentation of networking-related systems. Incorporates techniques from previous related courses. Emphasizes professional and ethical responsibilities; the need to stay current on technology; and its global impact on economics, society, and the environment.

CS 179J Computer Architecture and Embedded Systems (4) Discussion, 1 hour; laboratory, 9 hours; laboratory, 3 hours. Prerequisite(s): CS 100, CS 111, CS 122A, and CS 161 with grades of “C-” or better. ENGR 180W; 3 additional upper-division units in Computer Science. Covers the planning, design, implementation, testing, and documentation of a computer architecture and embedded systems-related system. Incorporates techniques from previous related courses. Emphasizes professional and ethical responsibilities; the need to stay current on technology; and its global impact on economics, society, and the environment.

CS 179K Software Testing and Verification (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 141; consent of instructor. Independent reading in material not covered in course work. Normally taken in senior year. Total credit for CS 194 may not exceed 8 units.

CS 179N Graphics and Electronic Games (4) Discussion, 1 hour; laboratory, 9 hours. Prerequisite(s): CS 130 with a grade of “C-” or better; ENGR 180W; 8 additional upper-division units in Computer Science. Covers the planning, design, implementation, testing, and documentation of a graphics- or electronic game-related system. Incorporates techniques presented in previous related courses. Emphasizes professional and ethical responsibilities; the need to stay current on technology; and its global impact on economics, society, and the environment.

CS 180 Introduction to Software Engineering (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 100. A study of software engineering techniques for the development, maintenance, and evolution of large software systems. Topics include requirements engineering; specification; implementation; debugging, testing, and quality assurance; reengineering; project management; software process; tools; and environments.

CS 181 Principles of Programming Languages (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 061, CS 100, CS 111, CS 150. Covers the principles of programming languages design. Includes the study and comparison of several programming languages, their features, and their implementations.

CS 182 Software Testing and Verification (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 100. A study of modern techniques to assess the quality of software artifacts through functional, performance, and reliability testing. Introduces black box and white box testing techniques. Covers the application of modern testing tools to software units, components, subsystems, and entire systems. Also covers verification as a complementary technique to testing.

CS 183 UNIX System Administration (4) Seminar, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 100. Explores the technical aspects of system administration on a Unix system, Apache, and MySQL. Includes managing system devices, operating system installation, communications, and networking.

CS 190 Special Studies (1-5) Individual study, 3-15 hours. Prerequisite(s): consent of instructor and department chair. Individual study to meet special curricular needs. Course is repeatable to a maximum of 9 units.

CS 191 Seminar in Research Topics in Computer Science and Engineering (1) Seminar, 1 hour. Prerequisite(s): upper division or graduate standing or consent of instructor. An introduction to the range of research topics and methods in Computer Science and Engineering and to the research opportunities available within the department. Graded Satisfactory (S) or No Credit (NC). Course is repeatable to a maximum of 3 units.

CS 193 Design Project (1-4) laboratory, 1-6 hours; scheduled research, 1-3 hours; individual study, 1-3 hours. Prerequisite(s): CS 141; consent of instructor. Individual hardware or software design project to include establishment of objectives and criteria, synthesis, analysis, implementation, testing, and documentation. Course is repeatable to a maximum of 8 units.

CS 194 Independent Reading (1-4) Prerequisite(s): consent of instructor. Independent reading in material not covered in course work. Normally taken in senior year. Total credit for CS 194 may not exceed 8 units.

CS 198-I Individual Internship in Computer Science (1-4) Internship, 3-12 hours. Prerequisite(s): upper-division standing; at least 12 units in Computer Science courses. An academic internship to provide the student with career experience as a computer scientist in a governmental, industrial, or research unit; or a faculty member in Computer Science. Each individual program must have the prior approval of
both supervisors and the Department chair. A final written report is required. Course is repeatable to a maximum of 8 units.

Graduate Courses

CS 201 Compiler Construction (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 152. Covers theory of parsing and translation. Also addresses compiler construction including lexical analysis, syntax analysis, code generation, and optimization. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 202 Advanced Operating Systems (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 153. Examines recent developments in operating systems. Also covers multiprocessing, parallel programming, time sharing, scheduling and resource allocation, and selected topics. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 203 Advanced Computer Architecture (4) Lecture, 3 hours; research, 3 hours. Prerequisite(s): CS 161. Covers contemporary computer systems architecture including stack computers, parallel computers, pipeline processing, database machines, and multiprocessor architecture. Includes evaluation of computer performance. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 204 Advanced Computer Networks (4) Lecture, 3 hours; consultation, 1 hour. Prerequisite(s): CS 014 with a grade of "C-" or better, CS 164. Covers advanced topics in computer networks, layering, Integrated Services Digital Networks (ISDN), and high-speed networks. Also covers performance models and analysis, distributed systems and databases, and case studies. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 206 Testing and Verification Techniques in Software Engineering (4) Lecture, 3 hours; individual study, 3 hours. Prerequisite(s): CS 141, CS 150, or equivalent; graduate standing. Introduces techniques to verify that software runtime behavior meets its specifications. Topics include model checking (safety, liveness, temporal logics, and abstraction); static and dynamic analysis (data flow analysis, control analysis, program slicing, and invariant detection); testing (test generation, suite reduction, and automated debugging); and automated debugging (fault location and visualization). May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 207 Advanced Programming Languages (4) Lecture, 3 hours; outside research, 1.5 hours; written work, 1.5 hours. Prerequisite(s): CS 152, CS 181, or equivalents. Introduces techniques for analyzing program semantics and correctness. Covers simply-typed lambda calculus as well as basic and advanced type systems. Presents axiomatic, operational, and denotational semantics. Explores program synthesis-language constructs and tools for specifying, reasoning, and verifying correctness properties. Includes safe memory accesses and safe concurrent programming or security. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 210 Scientific Computing (4) Lecture, 4 hours. Prerequisite(s): CS 012 or CS 012V; MATH 010A; MATH 031 or equivalent; or consent of instructor. Utilizes scientific computing in a specific computer science research area. Provides a foundation for pursuit of further studies of special topics in scientific computing. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 211 High Performance Computing (4) Lecture, 3 hours; research, 3 hours. Prerequisite(s): CS 161 or consent of instructor. Introduces performance optimization for sequential computer programs. Covers high performance computing on multicore shared memory computers and on distributed memory computing clusters. Also covers high performance scientific libraries and computing application development using pthreads. OpenMP and tasking interface (MPI) parallel file systems. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 213 Multiprocessor Architecture and Programming (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 203 or consent of instructor. Introduces multiprocessing architectures and CC-NuMa multiprocessors. Also covers heterogeneous multiprocessors and interconnection networks. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 215 Theory of Computation (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 150. Covers finite state grammar systems and languages, turing machines, and the design of automata; solvable and unsolvable problems; and theoretical limitations of computers. Also examines algorithmic complexity theory; polynomial reducibility; the classes P and NP; and correctness proofs. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 217 GPU Architecture and Parallel Programming (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 160 with a grade of "C-" or better or consent of instructor. Introduces the popular CUDA based parallel programming environments based on Nvidia GPUs. Covers the basic CUDA memory/threading models. Also covers the common data-parallel programming patterns needed to develop a high-performance parallel computing applications. Examines computational thinking, a parallel execution model, and parallel programming models. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 218 Design and Analysis of Algorithms (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141. A study of efficient data structures and algorithms for solving problems from a variety of areas such as sorting, searching, selection, linear algebra, graph theory, and computational geometry. Also covers worst-case vs. average-case analysis using recurrence relations, generating functions, upper and lower bounds, and other methods. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 220 Synthesis of Digital Systems (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141, CS 161. Covers formal methods and simulation of digital systems. Topics include synthesis at the system, behavioral, register-transfer, and logic levels; application-specific processors; simulation; and emerging systems and design technologies. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 223 Reconﬁgurable Computing (4) Lecture, 3 hours; written work, 3 hours. Prerequisite(s): CS 202 or CS 203; consent of instructor. Covers reconﬁgurable computing, a novel computational model that is fast becoming part of the mainstream in high-performance computing. Addresses architectures, software tools and compilers, programming models, and applications. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 229 Machine Learning (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141, STAT 160A. A study of supervised machine learning that emphasizes discriminative methods. Covers the areas of regression and classiﬁcation and includes linear methods, instance-based learning, neural networks, kernel machines, and additive models. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 230 Computer Graphics (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141 or CS 218, MATH 031 or MATH 131, graduate standing or consent of instructor. Covers the latest topics related to graphics and necessary fundamentals. Includes geometry representations, affine and perspective transforms; rendering with global illumination and other light models; shading and texture mapping; rasterization and anti-aliasing techniques; and hierarchical and keyframe animation. Also includes projects and/or in-depth programming assignments. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 231 Computer Animation (4) Lecture, 3 hours; research, 3 hours. Prerequisite(s): CS 130 or CS 230. Covers topics in computer animation including motion capture, inverse kinematics, and dynamic simulation. Also examines deformable systems and other natural phenomena, facial animation, high-level behavior control, creatures, emotions, and procedural techniques. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 233 Pen-Based Computing (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): graduate standing or consent of instructor; computer programming experience. Introduces computational techniques for pen-based user interfaces and fundamental issues such as ink segmentation, sketch parsing, and shape recognition. Explores the topic of sketch understanding, including reasoning about context and correcting errors. Also addresses issues such as building practical pen-based systems. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor. Cross-listed with ME 231.

CS 234 Computational Methods for Biomolecular Data (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 111; CS 141 or CS 218; STAT 155 or STAT 160A. A study of computational and statistical methods aimed at automated discovery, identifying, and classifying biomolecular data. Includes combinatorial algorithms for pattern discovery; hidden Markov models for sequence analysis; analysis of expression data; and prediction of the three-dimensional structure of RNA and proteins. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 235 Data Mining Techniques (4) Lecture, 3 hours; term paper, 1.5 hours; project, 1.5 hours. Prerequisite(s): CS 141, CS 166; CS 170 is recommended. Provides students with a broad background in the design and use of data mining algorithms and tools. Includes clustering, classification, association rules mining, time series clustering, and Web mining. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 236 Database Management Systems (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141, CS 153 or equivalent CS 166, or consent of instructor. Covers principles of file systems; architecture of database management systems; data models; and relational databases. Also examines logical and physical design of databases; hardware and software implementation of database systems; and distributed databases (e.g., query processing, concurrences, recovery). May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.
CS 237 Advanced Topics in Modeling and Simulation (4) Lecture, 3 hours; research, 3 hours. Prerequisite(s): CS 177. Covers formal computer simulation models such as Discrete Event Specified Models and differential equation models. Examines current developments in simulation languages. Also addresses integrated model development and its applications to complex, large-scale problems. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 238 Algorithmic Techniques in Computational Biology (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141 or CS 218. A study of fundamental algorithms for solving combinatorial or computational problems in molecular biology and genomics. Includes sequence alignment and multiple alignment, database search, gene and regulatory signal recognition, DNA sequence assembly, physical mapping, and reconstruction of evolutionary trees. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 239 Performance Evaluation of Computer Networks (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141 or CS 164 or CS 165. Discusses the theory and practice of distributed systems with a focus on recent developments in distributed systems. Includes middleware architectures; distributed process management and real-time scheduling; dependability; and group communication protocols. Also covers performance management; replication; large-scale peer-to-peer systems; Internet content delivery; and Web caching. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 240 Network Routing (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141 or CS 204 or CS 164. An in-depth study of routing in computer networks. Examines general principles and specific routing protocols and technologies. Topics include Internet, Asynchronous Transfer Mode (ATM), optical, wireless, and ad hoc networks. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 241 Advanced Topics in Network Measurements and Security (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 164 or equivalent. Introduces measuring and building real network systems. Includes hands-on measurement studies and tools. Covers fundamental mathematical and statistical tools; exposure to implementation studies and techniques; principles of network architectures; and challenges in building and conducting measurements. Explores measurements and modeling of wireless, ad hoc, sensor, and cellular networks. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 242 Information Retrieval and Web Search (4) Lecture, 3 hours; term paper, 1.5 hours; project, 1.5 hours per week. Prerequisite(s): CS 141, CS 166. Introduces Information Retrieval (IR) principles and techniques for indexing and searching document collections with special emphasis on Web search. Includes text preprocessing, ranking algorithms, search in social networks, search evaluation, and search engines scalability. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 245 Software Evolution (4) Lecture, 3 hours; research, 3 hours. Prerequisite(s): CS 180 or graduate; graduate standing. Covers the principles, tools, and techniques for disciplined software evolution. Includes migration strategies, change patterns, software maintenance, legacy system reengineering, reverse engineering, program understanding, middleware source code analysis, software visualization, and program transformation tools. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 246 Advanced Verification Techniques in Software Engineering (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 111, CS 141, CS 150, or equivalents or consent of instructor. A study of advanced techniques to specify and examine the correctness of complex systems and software. Focuses on concurrent and distributed behavior, formal description languages, temporal logics, model checking and symbolic model checking, partial order reduction, and the use of verification tools. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 253 Distributed Systems (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 153. Integrates the theory and practice of distributed systems with a focus on recent developments in distributed systems. Includes middleware architectures; distributed process management and real-time scheduling; dependability; and group communication protocols. Also covers performance management; replication; large-scale peer-to-peer systems; Internet content delivery; and Web caching. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 255 Computer Security (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 153 or CS 164 or CS 165. Discusses the theoretical and practical issues arising in the context of computer systems security and the principles underlying the design of secure computing environments. Topics include cryptographic techniques; password security; network security, intrusion detection, attacks and their countermeasures, and systems design. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 257 Wireless Networks and Mobile Computing (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141; CS 164 or CS 204. Introduces basic and advanced concepts of wireless networks and mobile computing. Covers both wireless cellular and ad hoc networks. Includes protocols for medium access control, resource allocation, and routing, as well as transport requirements for the wireless environment. Also covers standards, Bluetooth, and the IEEE 802.11 for wireless local area networks. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 260 Seminar in Computer Science (4) Seminar, 4 hours. Prerequisite(s): consent of instructor. Covers current research topics in computer science. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor. Course is repeatable to a maximum of 8 units.

CS 261 Seminar in Artificial Intelligence and the Design of Expert Systems (4) Seminar, 4 hours. Prerequisite(s): graduate standing or consent of instructor. A review of recent research topics in the fields of artificial intelligence and logic programming. Emphasizes expert systems, automatic reasoning, and knowledge representation. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor. Course is repeatable to a maximum of 8 units.

CS 262 Algorithms and Data Structures (4) Seminar, 4 hours. Prerequisite(s): CS 215, CS 218; or consent of instructor. Selected topics in theoretical computer science. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor. Course is repeatable to a maximum of 8 units.

CS 263 Seminar in Distributed Systems (4) Seminar, 4 hours. Prerequisite(s): graduate standing; CS 153 or previous operating systems course. Introduces the fundamental topics in distributed computer systems. Topics include distributed systems, replicated data, load management, and distributed shared memory. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor. Course is repeatable to a maximum of 8 units.

CS 267 Seminar in Databases (4) Seminar, 4 hours. Prerequisite(s): CS 236 or consent of instructor. Focusses on recent research and development issues in the database area. Includes object-oriented databases, heterogeneous databases, parallel databases, benchmarks, transaction processing, query optimization, and performance evaluation. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor. Course is repeatable to a maximum of 8 units.

CS 269 Software and Hardware Engineering of Embedded Systems (4) Seminar, 4 hours. Prerequisite(s): CS 120A/EE 120A; consent of instructor. Presents state-of-the-art software and hardware design techniques for embedded computing systems. Topics include specification models, languages, simulation, partitioning algorithms, estimation methods, model refinement, and design methodology. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor. Course is repeatable to a maximum of 8 units.

CS 270 Special Topics in Advanced Computer Science (2) Seminar, 2 hours. Prerequisite(s): consent of instructor. Involves presentations and discussions by faculty and students that focus on new research in computer science. Graded Satisfactory (S) or No Credit (NC). Course is repeatable.

CS 272 Probabilistic Models for Artificial Intelligence (4) Lecture, 3 hours; written work, 3 hours. Prerequisite(s): CS 141 or STAT 160. Covers aspects of reasoning about probability distributions in complex domains. Focuses on graphical models and their extensions such as Bayesian networks, Markov networks, hidden Markov models, and dynamic Bayesian networks. Topics include algorithms for probabilistic inference, learning models from data, and decision making. May be taken Satisfactory (S) or No Credit (NC) with consent of instructor and graduate advisor.

CS 287 Colloquium in Computer Science (1) Colloquium, 1 hour. Prerequisite(s): graduate standing. Lecture on current research topics in computer science and topics relating to professional development presented by faculty members and visitors. Graded Satisfactory (S) or No Credit (NC). Course is repeatable.

CS 290 Directed Studies (1-6) Individual study, 3-18 hours. Prerequisite(s): consent of instructor. Special studies in computer science. Graded Satisfactory (S) or No Credit (NC). Course is repeatable to a maximum of 12 units.

CS 297 Directed Research (1-6) Outside research, 3-18 hours. Prerequisite(s): graduate standing; consent of instructor. Directed research on selected projects in computer science under the sponsorship of assigned faculty members. Graded Satisfactory (S) or No Credit (NC). Course is repeatable to a maximum of 72 units.

CS 298-I Individual Internship (1-12) Written work, 1-12 hours; internship, 2-24 hours. Prerequisite(s): enrollment limited to teaching assistants and associates in Computer Science. Includes fieldwork with an approved professional individual or organization and academic work under the direction of a faculty member. A final written report is required. Graded Satisfactory (S) or No Credit (NC). Course is repeatable to a maximum of 16 units.

CS 299 Research for Thesis or Dissertation (1-12) Outside research, 3-36 hours. Prerequisite(s): graduate standing; consent of instructor. Research in computer science under the direction of a faculty member. To be included as part of the thesis or dissertation. Graded Satisfactory (S) or No Credit (NC). Course is repeatable.

Professional Courses

CS 302 Apprentice Teaching (1) Activity, 3 hours. Prerequisite(s): enrollment limited to teaching assistants and associates in Computer Science.