EE 175A and EE 175B. The selection of the remaining technical electives must be planned, in consultation with a faculty advisor, to include at least one coherent sequence of two classes from either Computer Science and Engineering or Electrical Engineering. The technical electives must be distinct from those used to satisfy the upper-division requirements specified in items a) and b) above.

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, and exceptions to Electrical Engineering course requirements must be approved by the Electrical Engineering undergraduate advisor or chair. Exceptions to other requirements require the approval of the undergraduate advisors or chairs of both departments.

Visit the Student Affairs Office in the College of Engineering or **student.engr.ucr.edu** for a sample program.

Computer Science and Engineering

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

Laxmi N. Bhuyan, Ph.D., Chair
Department Office,
351 Engineering Building Unit II
(951) 827-5639; **www1.cs.ucr.edu**

Professors
Laxmi N. Bhuyan, Ph.D.
Marek Chrobak, Ph.D.
Gianfranco Ciardo, Ph.D.
Michalis Faloutsos, Ph.D.
Rajiv Gupta, Ph.D.
Tao Jiang, Ph.D. *President's Chair*
Eamonn Keogh, Ph.D.
Srikanth Krishnamurthy, Ph.D.
Mart L. Molle, Ph.D.
Walid Najjar, 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.
Lawrence L. Larmore, Ph.D.
Thomas H. Payne, Ph.D.
Teodor C. Przymusinski, Ph.D.

Associate Professors
Stefano Lonardi, Ph.D.
Christian Shelton, Ph.D.
Victor Zordan, Ph.D.

Assistant Professors
Philip Brisk, Ph.D.
Harsha Madhyastha, Ph.D.
Iulian Neamtiu, Ph.D

**

Adjunct Professor
Douglas Tolbert, Ph.D.

Cooperating Faculty
Alexander Balandin, Ph.D. (Electrical Engineering)
Matthew J. Barth, Ph.D. (Electrical Engineering)
Bir Bhanu, Ph.D. (Electrical Engineering)
Ilya Dumer, Ph.D. (Electrical Engineering)
Lawrence H. Harper, Ph.D. (Mathematics)
Roger Lake, Ph.D. (Electrical Engineering)
Michel L. Lapidus, Ph.D. (Mathematics
Anastacios Mourikis, Ph.D., (Electrical Engineering)
Erik Rolland, Ph.D. (Accounting and Information Systems)
Amit Roy Chowdhury, Ph.D. (Electrical Engineering)
Thomas Stahovich, Ph.D. (Mechanical Engineering)
Xiang-Dong "Sheldon" Tan, Ph.D. (Electrical Engineering)
Zhengyuan "Daniel" Xu, Ph.D. (Electrical 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 engineering, networks, databases, graphics, algorithms, security, system analysis, and embedded systems.

The **Computer Engineering** major stresses the study of core computer science and electrical engineering topics. It prepares students for careers in the design of complex systems involving computer hardware, computer software, electronics and electrical signals for communication, networking, desktop computing, and embedded computing. The major is offered jointly by the Departments of Computer Science and Engineering, and Electrical Engineering. See Computer Engineering in this catalog.

The objective of the B.S. degree program in Computer Science is to prepare graduates for professional practice in both the private and public sectors and for life-long learning, including the option for graduate degrees, by providing them with:

- Background: the necessary technical competencies, including knowledge of scientific principles and skill at rigorous analysis and creative design
- Breadth: a broad education that includes knowledge of current issues and trends in society and technology
- Professionalism: professional attitudes and ethics and skills for clear communication and responsible teamwork
- Learning environment: a learning environment that is rigorous, challenging, open, and supportive

The **Business Informatics** major covers the core of computer science and basic business and management topics. It prepares students for careers in design and management of computer and information systems, system and network administration, and e-commerce. It is also useful for careers that apply information technology to support business processes.

The objective of the B.S. degree program in Business Informatics is to prepare graduates for professional practice in both the private and public sectors and for life-long learning, including the option for graduate degrees, by providing them with:

- Background: the necessary technical competencies, including knowledge of scientific principles and skill at rigorous analysis and creative design
- Breadth: a broad education that includes knowledge of current issues and trends in society and technology
- Professionalism: professional attitudes and ethics and skills for clear communication and responsible teamwork
- Learning environment: a learning environment that is rigorous, challenging, open, and supportive

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 (56 units)
   a) ENGR 001-I
   b) CS 010, CS 012 or CS 013, CS 014, CS 061

c) CS 011/MATH 011

d) MATH 008B or MATH 009A, MATH 009B, MATH 009C, MATH 010A

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 (90 units minimum)

a) ENGR 101-I

b) CS 100, CS 141, CS 150, CS 152, CS 153, CS 161, CS 161L, CS 179 (E-Z)

c) CS 120A/EE 120A, CS 120B/EE 120B

d) CS 111/MATH 111

e) ENGR 180W

f) MATH 113

g) STAT 155

h) Two courses from MATH 046, MATH 120, MATH 126, PHIL 124

i) At least 24 units of technical electives to be chosen from an approved list of courses which currently includes CS 100, 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 170, CS 177, CS 179 (E-Z) (4 units maximum), CS 180, CS 181, CS 183, CS 193 (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.

Visit the Student Affairs Office in the College of Engineering or **student.engr.ucr.edu** for a sample program.

Business Informatics Major

1. Lower-division requirements (51 units)

a) ENGR 001M

b) BUS 020

c) CS 010, CS 012 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

2. Upper-division requirements (98 units)

a) ENGR 101M

b) BUS 101, BUS 103, BUS 104/STAT 104, BUS 106/ECON 134

c) CS 100, CS 141, CS 153, CS 164, CS 165, CS 166, CS 180

d) CS 111/MATH 111

e) ENGR 180W

f) MATH 113

g) SOC 150

h) STAT 155

i) Twelve (12) units of upper-division Computer Science technical electives, which must be distinct from the above major requirements. These 12 units may be chosen from those courses listed as upper-division requirements or technical electives for the Computer Science major. At least two courses must be in the Department of Computer Science and Engineering.

j) Twenty (20) units of Business Administration technical electives, including at least 8 units of management information systems courses. These 20 units must be distinct from the above major requirements and may be chosen from any of the available Business Administration courses.

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, and exceptions to the Business Administration course requirements must be approved by the Graduate School of Management dean. Exceptions to other requirements require the approval of both the Department of Computer Science and Engineering and the Graduate School of Management.

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. Prerequisite courses: CS 010, CS 012 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/MATH 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.

Combined B.S. + M.S. Five-Year Program The college offers a combined B.S. + M.S. program in Computer Science designed to lead to a Bachelor of Science degree as well as a Master of Science degree in five years. Applicants for this program must have a high school GPA above 3.6, a combined SAT Reasoning score above 1950 (or ACT plus Writing equivalent), complete the Entry Level Writing Requirement before matriculation, and have sufficient mathematics preparation to enroll in calculus in their first quarter as freshmen.

Interested students who are entering their junior year should check with their academic advisor for information on eligibility and other details.

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 150, CS 152, CS 153, CS 161/CS 161L

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 203A or CS 220.
2. Theoretical foundations: CS 215 or CS 218.

3. Software and systems: CS 201 or CS 202.

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 215*, CS 218*, CS 234, CS 238

B. Computer Architecture, Embedded Systems, and CAD: CS 203A*, CS 203B, CS 213, CS 220*, CS 223, EE 213

C. Databases, Data Mining, and Machine Learning: CS 205*, CS 235*, CS 229, CS 236*, CS 272

D. Operating Systems and Distributed Systems: CS 202*, CS 253, CS 255*, CS 237

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*

G. Computer Graphics and Human-Computer Interaction: CS 230*, CS 231*, CS 233, ME 230, ME 231

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. Petitions for an attendance waiver in a specific quarter will be granted on a case-by-case basis. Students not waived from course attendance must enroll in CS 287 each quarter and receive a grade of "S."

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.

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)

a. Project Option. A student pursuing the M.S. degree, non-thesis option, may include up to 4 units of Directed Studies (CS 290) 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.

b. 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, according to one of the following options. 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. Project Option Students must complete a research project under the guidance of a faculty member. This project will require a written report and will be presented to a committee of at least two faculty members in an oral examination.

b. 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 must be presented to and approved by a committee of at least three faculty members.

The normative time for the completion of a M.S. in CS is 2 years.

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 48 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 B- and an overall core course GPA of at least 3.2.

2. Depth Requirement (12 units). Choose three 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, CS 261, CS 262, CS 263, CS 267, CS 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 the Ph.D. course requirements described above, and passed both the written qualifying examination and oral qualifying examination 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.

The Written Qualifying Exam. The written qualifying examination consists of a high-quality paper, solely authored by the student. This can be either a research paper containing an original contribution or a focused critical survey paper. The paper should demonstrate that the student understands and can integrate and communicate ideas clearly and concisely and should be approximately 10 pages, single-spaced. The organization and writing style of the paper should be suitable for submission to a first-rate technical conference or journal. It must represent work that the student did as a graduate student at UCR. Any contributions that are not the student's own, including those of the major professor, must be explicitly acknowledged in detail. The paper must be approved by the student's major professor prior to submission and must have a cover page with the advisor's signature, indicating approval. After submission, the paper is reviewed and must be approved by at least two other members of the faculty selected by the Department's Graduate Committee. The normative time for taking the Written Exam is the first quarter of the second year of graduate studies at UCR. The student must complete this requirement in no more than two attempts.

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 evaluates the merits of the work and the student's aptitude for research. The work must represent significant progress towards original and publishable research. A written report summarizing the oral presentation must be submitted to the Qualifying Committee at least a week before the exam. 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.

Milestone II: Dissertation Proposal Examination After advancement to candidacy, the student prepares a dissertation proposal that describes the dissertation topic, summarizes the relevant background literature, and presents a comprehensive research plan for the doctoral dissertation. The Dissertation Proposal Examination evaluates appropriateness of the research topic and the feasibility of the research plan. It also establishes a realistic timeline for the completion of the Dissertation. The Dissertation Committee administers this exam. The normative time for the Dissertation Proposal Exam is by the end of the third year. The Dissertation Proposal exam must be taken at least six months prior to the Final Doctoral Examination.

Milestone III: Final Doctoral Examination The student is required to write a dissertation in accordance with the Graduate Division requirements and may be required to defend it in a public oral final doctoral examination to the Dissertation Committee. After a satisfactory performance on the final doctoral examination, the Dissertation Committee recommends granting the PhD degree. The student's research and the dissertation must both meet the highest standards of originality and scholarship.

The normative time for the completion of a Ph.D. in CS is five years.

Lower-Division Courses

CS 005. Introduction to Computer Programming (4) Lecture, 3 hours; laboratory, 3 hours. An introduction to computer programming for nonengineering and nonscience majors and for students considering taking CS 010 but needing additional preparation. Topics include the history of computing, basic computer operation, the notion of an algorithm, and programming constructs such as variables, expressions, input/output, branches, loops, functions, parameters, arrays, and strings. Credit is not awarded for CS 005 if it has already been awarded for CS 010.

CS 006. Effective Use of the World Wide Web (4) Lecture, 3 hours; laboratory, 3 hours. A detailed, nontechnical introduction to the Internet, covering Web tools, e-communities, e-commerce, power searching, and verification of information, privacy, and other legal and societal issues.

CS 008. Introduction to Computing (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): none. Includes operating system basics (Windows and Unix), word processing, spreadsheets, databases (e.g., Access), E-mail, the Internet, and the World Wide Web. Designed for students not majoring in computer science, engineering, mathematics, or science. Credit is not awarded for CS 008 if it has already been awarded for CS 010.

CS 010. Introduction to Computer Science for Science, Mathematics, and Engineering I (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): a UCR 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. Credit is not awarded for CS 010 if it has already been awarded for 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 MATH 009B or MATH 09HB. Introduction to basic concepts of discrete mathematics with emphasis on applications to computer science. Topics include prepositional and predicate calculi, 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 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 013.

CS 013. Introductory Computer Science for Engineering Majors (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 010 with a grade of "C" or better; familiarity with C or C++ language. A course corresponding to CS 012, but tailored to engineering majors. 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 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 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; dictionaries including binary search trees and hashing; priority queues (heaps); introductory analysis of algorithms; sorting algorithms; and object-oriented programming including abstract data types, inheritance, and polymorphism. Also covers solving complex problems through structured software development.

CS 021. Introduction to UNIX (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 008 or CS 010. A comprehensive introduction to fundamental UNIX principles, architecture, and applications. Covers command-line utilities, piping, redirection, filters, basic shell scripting, file system architecture and permissions, and tools for software compilation, debugging, and version control. Topics are relevant to and enhance students' work in most lower- and upper-division Computer Science courses.

CS 030. Introduction to Computational Science and Engineering (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): MATH 009C (may be taken concurrently); consent of instructor if credit has been awarded for CS 010. Examines fundamental programming concepts using the Matlab language, including problem decomposition, control structures, elementary data structures, file input/output, graphics, and code libraries. Focuses on applications problems in engineering and science, such as numerical equation solvers; matrix operations; searching and sorting; and data analysis. Emphasizes good programming style and computational efficiency.

CS 049 (E-Z). Language Laboratory (2) For hours and prerequisites, see segment descriptions. Hands-on, directed exposure to a specific programming language and associated development tools in a laboratory setting. Focus is on exercises and practical applications

Graded Satisfactory (S) or No Credit (NC). Each segment is repeatable as topics change to a maximum of 12 units.

CS 049E. Introductory C and C++ (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 010 or knowledge of programming or consent of instructor. Practical introduction to software development using C and C++ in a laboratory setting. Focus is on syntax, concepts, selections from the standard library, 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 049G. Advanced C++ (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 010 or knowledge of programming or consent of instructor. Exploration of selected advanced topics in C++ in a laboratory setting. Topics include advanced object-oriented techniques, standard template library, interface design, and advanced idiomatic development practices. Graded Satisfactory (S) or No Credit (NC). Segment is repeatable as topics change to a maximum of 12 units.

CS 049-I. C# (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 010 or knowledge of programming or consent of instructor. Practical exploration of software development using C# in a laboratory setting. Focus is on syntax, concepts, standard library, 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 049J. Introductory Java (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 010 or knowledge of programming or consent of instructor. Practical introduction to software development using Java in a laboratory setting. Focus is on syntax, concepts, standard, selections from the standard library, 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 049M. Matlab (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 010 or knowledge of programming or consent of instructor. Practical exploration of problem solving using Matlab in a laboratory setting. Focus is 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 049N. Hardware Description (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 010 or knowledge of programming or consent of instructor. Introduces hardware description languages (HDLs) used to design modern digital integrated circuits found in a wide variety of electronic devices. Topics include modeling of circuit structure, register transfers, high-level behavior and testbenches; HDL simulation models; use of synthesis tools; and tradeoffs among HDLs. Graded Satisfactory (S) or No Credit (NC). Segment is repeatable as topics change to a maximum of 12 units.

CS 049Q. Perl (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 010 or knowledge of programming or consent of instructor. Practical exploration of problem solving and software development using Perl in a laboratory setting. Focus is on syntax, concepts, and idiomatic use. Graded Satisfactory (S) or No Credit (NC). Segment is repeatable as topics change to a maximum of 12 units.

CS 049S. Bash (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 010 or knowledge of programming or consent of instructor. Practical exploration of problem solving using Bash scripting in a laboratory setting. Focus is on syntax, concepts, and idiomatic use. Graded Satisfactory (S) or No Credit (NC). Segment is repeatable as topics change to a maximum of 12 units.

CS 049Y. Python (2) Lecture, 1 hour; laboratory, 3 hours. Prerequisite(s): CS 005 or CS 010 or knowledge of programming or consent of instructor. Practical exploration of problem solving and software development using Python in a laboratory setting. Focus is on syntax, concepts, standard library, 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 061. Machine Organization and Assembly Language Programming (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 010 with a grade of "C" or better. An introduction to computer organization. Topics include number representation, combinational and sequential logic, computer instructions, memory organization, addressing modes, interrupt, input/output (I/O), assembly language programming, assemblers, and linkers.

Upper-Division Courses

CS 100. Software Construction (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 014. Covers the development and construction of software products. Topics include design, coding layout, and style; implementation strategies; quality attributes; prototyping, reuse, and components; debugging, testing, and performance; integration and maintenance; documentation; standards, analysis, and selection of tools and environment; and personal software processes.

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

CS 120A. Logic Design (5) Lecture, 3 hours; laboratory, 6 hours. Prerequisite(s): CS 061 with a grade of "C-" or better. Covers the design of digital systems. Topics include Boolean algebra; combinational and sequential logic design; design and use of arithmetic-logic units, carry-lookahead adders, multiplexors, decoders, comparators, multipliers, flip-flops, registers, and simple memories; state-machine design; and basic register-transfer level design. Interdisciplinary laboratories involve use of hardware description languages, synthesis tools, programmable logic, and significant hardware prototyping. Cross-listed with EE 120A.

CS 120B. Introduction to Embedded Systems (5) Lecture, 3 hours; laboratory, 6 hours. Prerequisite(s): CS 120A/EE 120A. Introduction to hardware and software design of digital computing systems embedded in electronic devices (such as digital cameras or portable video games). Topics include embedded processor programming, custom processor design, standard peripherals, memories, interfacing, and hardware/software tradeoffs. Interdisciplinary laboratory involves use of synthesis tools, programmable logic, and microcontrollers and development of working embedded systems. Cross-listed with EE 120B.

CS 122A. Intermediate Embedded and Real-Time Systems (5) Lecture, 3 hours; laboratory, 6 hours. Prerequisite(s): CS 012 or CS 013; CS 120B/EE 120B. Covers software and hardware design of embedded computing systems. Topics include hardware and software codesign, 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 advanced simulation, and debug 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 cosimulation and codesign, 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, CS 141, MATH 113 (MATH 113 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 111, MATH 113, 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 polytops, 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; laboratory, 3 hours. Prerequisite(s): CS 014 with a grade of "C-" or better; CS 111/MATH 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. Homework and programming assignments integrate knowledge of data structures, algorithms, and programming.

CS 143. Multimedia Technologies and Programming (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 010. Introduces multimedia technologies and programming techniques, multimedia hardware devices, authoring languages and environments, temporal and nontemporal media (interactivity in text, graphics, audio, video, and animation), applications, and trends. Requires a term project. Cross-listed with EE 143.

CS 145. Combinatorial Optimization Algorithms (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 141; MATH 113 or MATH 131. The 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. The Theory of Automata and Formal Languages (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 014; CS 111/MATH 111; MATH 009C or MATH 09HC. A study of formal languages, including 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; equivalence between the models, and an introduction to 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, complexity classes P and 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/MATH 111, CS 150. 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/MATH 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/MATH 111. A study of concurrent and parallel systems. Topics include modular structure and design, interprocess communication, synchronization, failures, persistence, and concurrency 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 120B/EE 120B; 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 120B/EE 120B; concurrent enrollment in CS 161. Students design and simulate 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 pipelining, 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, basic multiprocessor 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/MATH 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/MATH 111. Covers architecture of database management systems; relational, network, and hierarchical models; distributed database concepts; query languages; implementation issues; and privacy and security of the database.

CS 168. Introduction to Very Large Scale Integration (VLSI) Design (5) Lecture, 3 hours; laboratory, 6 hours. Prerequisite(s): CS 120A/EE 120A or consent of instructor. Basic electrical properties of metal-oxide-semi conductor (MOS) circuits. MOS circuit design processes. Basic circuit concepts. Subsystem design and layout. Aspects of system design. Memory, registers, and aspects of systems timing. Very large scale integration design.

CS 169. Mobile Wireless Networks (4) Lecture, 3 hours; laboratory, 2 hours; extra reading, 1 hour. Prerequisite(s): CS 164 or consent of instructor. Introduces the fundamentals of wireless and mobile networks. Covers wireless channel models; MAC protocols; and wireless network architectures. 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; laboratory, 3 hours. Prerequisite(s): CS 100, CS 111/MATH 111. An introduction to fundamental problems underlying the design of intelligent systems. Also covers one of the languages of artificial intelligence, such as Prolog or LISP. Includes brute force and heuristic search, problem solving, knowledge representation, predicate logic and logical interference, frames, semantic nets, natural language processing, and expert systems.

CS 171. Introduction to Expert Systems (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 170 or equivalent. Introduction to methodology of design and implementation of expert systems. Rule-based and frame-based expert systems. Knowledge acquisition and knowledge engineering. Design of expert system shells. Use of expert system shells to construct knowledge-based systems.

CS 177. Modeling and Simulation (4) Lecture, 3 hours; laboratory, 3 hours. Prerequisite(s): CS 100, CS 111/MATH 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 179E. Compilers (4) Discussion, 1 hour; laboratory, 9 hours. Prerequisite(s): CS 141 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 141 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. Prerequisite(s): CS 141 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 a network-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 179J. Computer Architecture and Embedded Systems (4) Discussion, 1 hour; laboratory, 9 hours. Prerequisite(s): CS 100, CS 111/MATH 111, CS 122A, and CS 161 with grades of "C-" or better or consent of instructor; 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 using 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 179K. Software Engineering (4) Discussion, 1 hour; laboratory, 9 hours. Prerequisite(s): CS 180; ENGR 180W; 8 additional upper-division units in Computer Science. Covers the planning, design, implementation, testing, and documentation of a software engineering-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 179M. Artificial Intelligence (4) Discussion, 1 hour; laboratory, 9 hours. Prerequisite(s): CS 100, CS 111/MATH 111, and CS 170 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 an artificial intelligence-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 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 using 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 014, CS 100. A study of software engineering techniques for the development, maintenance, and evolution of large software systems. Topics include requirements and specification; system design and 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/MATH 111, CS 150. Covers the principles of programming language design. Includes the study and comparison of several programming languages, their features, and their implementations.

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, including advanced Unix. 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 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 under the joint supervision of an off-campus sponsor and 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) by students advanced to candidacy for the Ph.D.

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 multiprogramming, parallel programming, time sharing, scheduling and resource allocation, and selected topics. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

CS 203A. 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) by students advanced to candidacy for the Ph.D.

CS 203B. Advanced Computer Architecture (4) Lecture, 3 hours; research, 3 hours. Prerequisite(s): CS 203A with a grade of "B" or better. Covers advanced topics in general-purpose computer architecture including instruction-level parallel architectures, as well as very-long-instruction-word, explicitly parallel instruction computing, and multithreaded architectures. Also covers dataflow machines and vector and single instruction multiple data architectures, including multimedia extensions. Also discusses network processors, multimedia processors, and advanced embedded processors. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

CS 204. Advanced Computer Networks (4) Lecture, 3 hours; consultation, 1 hour. Prerequisite(s): CS 014, 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) by students advanced to candidacy for the Ph.D.

CS 205. Artificial Intelligence (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 170 or equivalent. Examines knowledge representation and automated reasoning and their use in capturing common sense and expert knowledge. Also addresses predicate and nonmonotonic logics; resolution and term rewriting; reasoning under uncertainty; theorem provers; planning systems; and belief networks. Includes special topics in natural language processing, perception, logic programming, expert systems, and deductive databases. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

CS 206. Testing and Verification Techniques in Software Engineering (4) Lecture, 3 hours; individual study, 3 hours. Prerequisite(s): CS 141, CS 150, or equivalents; 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, concept analysis, program slicing, and invariant detection), testing (test generation, prioritization, suite reduction, and regression), and automated debugging (fault location and visualization). May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

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 the 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 programming-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 213. Parallel Processing Architectures (4) Lecture, 3 hours; discussion, 1 hour. Prerequisite(s): CS 161 or CS 203A. A study of parallel processing. Covers static and dynamic interconnection networks; shared memory multiprocessors; and cache coherence and synchronization. Also examines pre-fetching; memory management; message-passing architectures; workstation clusters; scheduling and mapping algorithms; and load balancing in Web servers. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

CS 215. Theory of Computation (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 150. Covers phrase structure grammars and languages; turing machines; relation of languages to 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) by students advanced to candidacy for the Ph.D.

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 and average-case analysis using recurrence relations, generating functions, upper and lower bounds, and other methods. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

CS 220. Synthesis of Digital Systems (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141, CS 161. Covers the synthesis and simulation of digital systems. Topics include synthesis at the system, behavioral, register-transfer, and logic levels; application-specific processors; simulation; and emerging system-on-a-chip design methodologies. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

CS 223. Reconfigurable Computing (4) Lecture, 3 hours; written work, 3 hours. Prerequisite(s): CS 202 or CS 203A; consent of instructor. Covers reconfig-

urable 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. Topics include linear methods, instance-based learning, neural networks, kernel machines, and additive models. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

CS 230. Computer Graphics (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141 or CS 218; MATH 113 or MATH 131; graduate standing or consent of instructor. Covers advanced 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) by students advanced to candidacy for the Ph.D.

CS 231. Computer Animation (4) Lecture, 3 hours; outside 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; creature evolution; and procedural techniques. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

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. Covers 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 related to 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/MATH 111, CS 141 or CS 218; STAT 155 or STAT 160A. A study of computational and statistical methods aimed at automatically analyzing, clustering, 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) by students advanced to candidacy for the Ph.D.

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) by students advanced to candidacy for the Ph.D.

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) by students advanced to candidacy for the Ph.D.

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) by students advanced to candidacy for the Ph.D.

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; bio-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) by students advanced to candidacy for the Ph.D.

CS 239. Performance Evaluation of Computer Networks (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 164. Offers models and analytical techniques for evaluating the performance of computer networks. Covers basic and intermediate queuing theory and queuing networks and their application to practical systems. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

CS 240. Network Routing (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 141 or CS 204; 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) by students advanced to candidacy for the Ph.D.

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 testbeds and conducting measurements. Explores measurements and modeling of wireline, ad hoc, sensor, and cellular networks. Course is repeatable as content changes to a maximum of 8 units.

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

CS 246. Advanced Verification Techniques in Software Engineering (4) Lecture, 3 hours; outside research, 3 hours. Prerequisite(s): CS 111/MATH 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) by students advanced to candidacy for the Ph.D.

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 distributed process management; replication; large-scale peer-to-peer systems; Internet content delivery; and Web caching. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

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 cryptography, security models, authentication protocols, network security, intrusion detection, attacks and their countermeasures, and secure systems design. May be taken Satisfactory (S) or No Credit (NC) by students advanced to candidacy for the Ph.D.

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 layer optimizations 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) by students advanced to candidacy for the Ph.D.

CS 260. Seminar in Computer Science (1-4) Seminar, 1-4 hours. Prerequisite(s): consent of instructor. Covers current research topics in computer science. Course is repeatable.

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 with a particular emphasis on expert systems, automated reasoning, and knowledge representation.

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. Course is repeatable.

CS 263. Seminar in Distributed Systems (4) Seminar, 4 hours. Prerequisite(s): graduate standing; CS 153 or previous operating systems course. A project-oriented course that introduces students to the fundamental topics in distributed computer systems and provides practical experience. Topics include distributed file systems, replicated data, load management, and distributed shared memory.

CS 267. Seminar in Databases (4) Seminar, 4 hours. Prerequisite(s): CS 236 or consent of instructor. Focuses on recent research and development issues in the database area such as object-oriented databases, heterogenous databases, parallel databases,

benchmarks, transaction processing, query optimization, and performance evaluation.

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.

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, STAT 160A. Covers methods for representing and 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) by students advanced to candidacy for the Ph.D.

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

CS 290. Directed Studies (1-6) Seminar, 1-6 hours. Prerequisite(s): consent of instructor. Research and special studies in computer science. Graded Satisfactory (S) or No Credit (NC). Course is repeatable.

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): graduate standing; consent of instructor. Individual apprenticeship 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 12 units.

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

Professional Courses

CS 301. Teaching Computer Science at the College Level (1) Seminar, 1 hour. Prerequisite(s): graduate standing. A program of weekly meetings and individual formative evaluation required of new Computer Science Teaching Assistants. Covers instructional methods and classroom/section activities most suitable for teaching Computer Science. Conducted by departmental faculty. Graded Satisfactory (S) or No Credit (NC). Course is repeatable.

CS 302. Apprentice Teaching (1-4) Seminar, 1-4 hours. Prerequisite(s): enrollment limited to teaching assistants and associates in Computer Science. Supervised teaching in upper- and lower-division Computer Science courses. Required each quarter of all Computer Science teaching assistants and associates. The course is intended to aid in the learning of effective teaching methods such as the handling of Computer Science discussion sections, preparation and grading of examinations, and student relations. Graded Satisfactory (S) or No Credit (NC).

Conservation Biology

Subject abbreviation: BLCN
College of Natural and Agricultural Sciences

Program Office, 1223 Pierce Hall
(951) 827-7294; **ccb.ucr.edu**

The major in Conservation Biology is not currently accepting new students. Students who are interested in this field should see the Conservation Biology track, in the Biological Sciences section of this catalog. For more information, contact the CNAS Undergraduate Office, (951) 827-7294.

Upper-Division Courses

BLCN 190. Special Studies (1-4) Individual study, 3-12 hours. Prerequisite(s): consent of instructor and Program Chair. To be taken as a means of meeting special curricular needs. Course content, style, requirements, and grading basis is selected in consultation with the instructor and Program Chair. Course is repeatable to a maximum of 12 units.

BLCN 197. Research for Undergraduates (1-2) Outside research, 3-6 hours. Prerequisite(s): sophomore, junior, or senior standing in Conservation Biology; consent of instructor and Program Chair. An introduction to research providing the opportunity, through reading and preliminary laboratory work, to develop a research project suitable for BLCN 199. Graded Satisfactory (S) or No Credit (NC). Course is repeatable to a maximum of 4 units.

BLCN 198-I. Individual Internship in Conservation Biology (2-4) Internship, 6-12 hours; consultation, 1 hour; outside reading, 2-4 hours. Prerequisite(s): upper-division standing in Conservation Biology. An off-campus practical experience in the public or private sector related to conservation biology that is conducted under the joint supervision of an off-campus sponsor and a faculty mentor from the Conservation Biology Program. A written report on the internship is required. Graded Satisfactory (S) or No Credit (NC). Course is repeatable to a maximum of 12 units.

BLCN 199. Senior Research (1-4) Laboratory, 3-12 hours. Prerequisite(s): junior or senior standing in Conservation Biology; consent of instructor and Program Chair. BLCN 197 is recommended. Research in conservation biology performed under the supervision of a faculty member in the Conservation Biology Program. A written research report is required. Graded Satisfactory (S) or No Credit (NC). Course is repeatable to a maximum of 12 units.

Creative Writing

Subject abbreviation: CRWT
College of Humanities, Arts, and Social Sciences

Tom Lutz, Ph.D., Chair
Department Office, 4146 CHASS INTS
(951) 827-5569; **creativewriting.ucr.edu**

Professors
Christopher Abani, Ph.D.
Christopher Buckley, M.F.A.
Mike Davis, C.Phil.
Stephanie Hammer, Ph.D.
Juan Felipe Herrera, M.F.A.
Tom Lutz, Ph.D.
Maurya Simon, M.F.A.
Susan C. Straight, M.F.A.

Professors Emeritus
Steve Minot, Ph.D.
Eliud Martínez, Ph.D.

Associate Professors
Reza Aslan, Ph.D.
Michael Jayme, M.F.A.
Laila Lalami, Ph.D.

Assistant Professors
Claire Hoffman, M.A., M.S.J.
Goldberry Long, M.F.A.
Andrew Winer, M.F.A.

Major

The Creative Writing major offers a series of workshop courses in poetry, fiction, playwriting, screenwriting, and nonfiction as well as reading courses in poetry and fiction presented from a writer's point of view. They are taught for the most part by poets, fiction writers, and playwrights.

The writing courses are taught as workshops, so that the subject matter (the students' stories, poems, and plays) is different each time the course is offered.

Incoming freshmen and transfer students can apply for a Chancellor's Performance Award, for up to $4,500. Contact the department office for more information.

University Requirements

See Undergraduate Studies section.

College Requirements

See College of Humanities, Arts, and Social Sciences, Colleges and Programs section.

Major Requirements

The major requirements for the B.A. degree in Creative Writing are as follows:

Prerequisite courses: CRWT 056 or equivalent, and ENGL 001A or equivalent.

1. Lower-division requirements (20 units; five courses)

   Two Creative Writing survey courses from CRWT 046A, CRWT 046B, or CRWT 046C

   and