

ates of the B.S. program that wish to seek immediate employment are prepared for a wide range of computer science positions in industry and government. Outstanding graduates interested in highly technical careers, research, and/or academia, might consider furthering their education in graduate school.

The primary computer science departmental emphasis is on problem solving using computer program design, analysis and implementation, with both a theoretical foundation and a practical component.

Program Outcomes for Undergraduate Programs

The program enables students to achieve, by the time of graduation:

1. An ability to apply knowledge of computing and mathematics appropriate to computer science.
2. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
3. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
4. An ability to function effectively on teams to accomplish a common goal.
5. An understanding of professional, ethical, and social responsibilities.
6. An ability to communicate effectively.
7. An ability to analyze the impact of computing on individuals, organizations, and society, including ethical, legal, security, and global policy issue.
8. Recognition of the need for and an ability to engage in continuing professional development.
9. An ability to use current techniques, skills, and tools necessary for computing practice.
10. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the trade-offs involved in design choices.
11. An ability to apply design and development principles in the construction of software systems of varying complexity.

Admission to the Major

Students interested in computer science who apply to UCSB should declare the computer science major when they apply. Computer science majors have priority when registering for all Computer Science Courses. UCSB students can petition for a change-of-major into the Department of Computer Science once the minimum requirements are completed.

Students applying for major status in the BS program who have completed more than 105 units will not be considered for a change of major.

Students may petition for a change-of-major once these requirements are met:

1. A cumulative grade point average of at least 3.0

2. Satisfactory completion of Computer Science 16, 24, and 40 with a cumulative GPA of 3.2 or higher; First takes only
3. Satisfactory completion of Math 3A, 3B, 4A, and 4B with a cumulative GPA of 3.0 or higher; First takes only

Students admitted to the computer science major are responsible for satisfying major requirements in effect when they declare their major. Upper and lower division courses required for the major that are offered by the Department of Computer Science or any other department must be taken for letter grades.

The selection process is highly competitive and these milestones are minimum requirements for consideration, achieving them does not guarantee admission to the Computer Science major. Any petitions denied will be automatically considered a second time in the next quarter. Petitions denied a second time will not be reconsidered.

More information can be found at <http://cs.ucsb.edu/undergraduate/admissions/>.

Undergraduate Program

Bachelor of Science—Computer Science

A minimum of 184 units is required for graduation. A complete list of requirements for the major can be found on page 50. Schedules should be planned to meet both General Education and major requirements.

Students with no previous programming background should take CMPSC 8 before taking CMPSC 16. CMPSC 8 is not included in the list of preparation for the major courses but may be counted as a free elective.

Bachelor of Science—Computer Engineering

This major is offered jointly by the Department of Computer Science and the Department of Electrical and Computer Engineering. For information about this major, see page 25.

Computer Science Courses

LOWER DIVISION

4. Computer Science Boot Camp

(4) KOC

NOT open to CMPSC or CMPEN Majors.

An introduction to computational thinking, computing, data management, and problem solving using computers, for non-majors. Topics include coding basics, representing code and data using a computer, and applications of computing that are important to society.

8. Introduction to Computer Science

(4) KHARITONOVA, MIRZA, MATNI

Not open for credit to students who have completed Computer Science 16 or Engineering 3.

Legal repeat for CMPSC 5AA-ZZ.

Introduction to computer program development

for students with little to no programming experience. Basic programming concepts, variables and expressions, data and control structures, algorithms, debugging, program design, and documentation.

16. Problem Solving with Computers I

(4) KHARITONOVA, MIRZA

Prerequisite: Math 3A with a grade of C or better (may be taken concurrently), Computer Science 8 or Engineering 3 or Electrical and Computer Engineering 3 with a grade of C or better, another university-level intro to programming course, or significant prior programming experience.

Legal repeat of CMPSC 10.

Fundamental building blocks for solving problems using computers. Topics include basic computer organization and programming constructs: memory CPU, binary arithmetic, variables, expressions, statements, conditionals, iteration, functions, parameters, recursion, primitive and composite data types, and basic operating system and debugging tool.

24. Problem Solving with Computers II

(4) AGRAWAL, MIRZA

Prerequisite: Computer Science 16 with a grade of C or better; and Math 3B (may be taken concurrently).

Not open for credit to students who have completed Computer Science 20.

Intermediate building blocks for solving problems using computers. Topics include intermediate object-oriented programming, data structures, object-oriented design, algorithms for manipulating these data structures and their run-time analyses. Data structures introduced include stacks, queues, lists, trees, and sets.

32. Object Oriented Design and Implementation

(4) WANG, R.

Prerequisite: Computer Science 24 with a grade of C or better.

Computer Science 32 is a legal repeat for Computer Science 60.

Advanced topics in object-oriented computing. Topics include encapsulation, data hiding, inheritance, polymorphism, compilation, linking and loading, memory management, and debugging; recent advances in design and development tools, practices, libraries, and operating system support.

40. Foundations of Computer Science

(5) VAN DAM, SU

Prerequisites: Computer Science 16 with a grade of C or better; and Mathematics 4A with a grade of C or better.

Introduction to the theoretical underpinnings of computer science. Topics include propositional predicate logic, set theory, functions and relations, counting, mathematical induction and recursion (generating functions).

48. Computer Science Project

(4) CONRAD, KRINTZ

Prerequisite: Computer Science 32 with a grade of C or better.

Team-based project development. Topics include software engineering and professional development practices, interface design, advanced library support; techniques for team-oriented design and development, testing and test-driven development, and software reliability and robustness. Students present and demonstrate their final projects.

56. Advanced Applications Programming

(4) CONRAD

Prerequisite: Computer Science 24 and 32 with a grade of C or better.

Not open for credit to students who have completed Computer Science 20

Advanced application programming using a highlevel, virtual-machine-based language. Topics include generic programming, exception handling, programming language implementation; automatic memory management, and application development, management, and maintenance tools; event handling, concurrency and threading, and advanced library use.

64. Computer Organization and Logic Design

(4) MATNI

Prerequisite: Computer Science 16 with a grade of C or better; and Mathematics 3C or 4A with a grade of C or better.

Not open for credit to students who have completed ECE 15 or ECE 15B or Computer Science 30. Course counts as a legal repeat of CMPSC 30.

Assembly language programming and advanced computer organization; Digital logic design topics including gates, combinational circuits, flip-flops, and the design and analysis of sequential circuits.

95AA-ZZ. Undergraduate Seminar in Computer Science

(1-4) STAFF

Prerequisites: Open to pre-computer science and pre-computer engineering majors only; consent of instructor.

Seminars on introductory topics in computer science. These seminars provide an overview of the history, technology, applications, and impact in various areas of computer science, including: A. Foundations, B. Software Systems, C. Programming languages and software engineering, D. Information management, E. Architecture, F. Networking, G. Security, H. Scientific computing, I. Intelligent and interactive systems, J. History, N. General.

99. Independent Studies in Computer Science

(1-4) STAFF

Must have a minimum 3.0 grade point average. May be repeated. Students are limited to 5 units per quarter and 30 units total in all 99/198/199 courses combined.

Independent studies in computer science for advanced students.

UPPER DIVISION

111. Introduction to Computational Science

(4) GILBERT, MATNI

Prerequisite: Mathematics 5A or 4B with a grade of C or better; Mathematics 5B or 6A with a grade of C or better; Computer Science 24 with a grade of C or better.

Not open for credit to students who have completed Computer Science 110A.

Introduction to computational science, emphasizing basic numerical algorithms and the informed use of mathematical software. Matrix computation, systems of linear and nonlinear equations, interpolation and zero finding, differential equations, numerical integration. Students learn and use the Matlab language.

130A. Data Structures and Algorithms I

(4) EL ABBADI, SINGH, SURI

Prerequisites: Computer Science 40 with a grade of C or better; Computer Science 32 with a grade of C or better; PSTAT 120A or ECE 139; open to computer science, computer engineering, and electrical engineering majors only.

The study of data structures and their applications. Correctness proofs and techniques for the design of correct programs. Internal and external searching. Hashing and height balanced trees. Analysis of sorting algorithms. Memory management. Graph traversal techniques and their applications.

130B. Data Structures and Algorithms II

(4) LOKSHTANOV, SINGH, SURI

Prerequisite: Computer Science 130A.

Design and analysis of computer algorithms. Correctness proofs and solution of recurrence relations. Design techniques: divide and conquer, greedy strategies, dynamic programming, branch and bound, backtracking, and local search. Applications of techniques to problems from several disciplines. NP - completeness.

138. Automata and Formal Languages

(4) EGECIOGLU

Prerequisite: Computer Science 40 with a grade of C or better; open to computer science and computer engineering majors only.

Formal languages; finite automata and regular expressions; properties of regular languages; pushdown automata and context-free grammars; properties of context-free languages; introduction to computability and unsolvability. Introduction to Turing machines and computational complexity.

140. Parallel Scientific Computing

(4) YANG, T., GILBERT

Prerequisite: Mathematics 4B or 5A with a grade of C or better; Mathematics 6A or 5B with a grade of C or better; Computer Science 130A.

Not open for credit to students who have completed Computer Science 110B.

Fundamentals of high performance computing and parallel algorithm design for numerical computation. Topics include parallel architecture and clusters, parallel programming with message-passing libraries and threads, program parallelization methodologies, parallel performance evaluation and optimization, parallel numerical algorithms and applications with different performance trade-offs.

153A. Hardware/Software Interface

(4) KRINTZ, BREWER

Prerequisite: Upper-division standing in computer science, computer engineering, or electrical engineering.

Same course as ECE 153A.

Issues in interfacing computing systems and software to practical I/O interfaces. Rapid response, real-time events and management of tasks, threads, and scheduling required for efficient design of embedded software and systems is discussed. Techniques for highly constrained systems.

154. Computer Architecture

(4) KOC, MATNI

Prerequisite: Computer Science 32 with a grade of C or better, Computer Science 48 with a grade of C or better, and Computer Science 64 with a grade of C or better.

Not open for credit to students who have received credit for ECE 154, ECE 154A, or ECE 154B.

Introduction to the architecture of computer systems. Topics include: central processing units, memory systems, channels and controllers, peripheral devices, interrupt systems, software versus hardware trade-offs.

160. Translation of Programming Languages

(4) DING, HARDEKOPF

Prerequisite: Computer Science 64 or Electrical Engineering 154 or Electrical Engineering 154A; Computer Science 130A; and Computer Science 138; open to computer science and computer engineering majors only.

Study of the structure of compilers. Topics include: lexical analysis; syntax analysis including LL and LR parsers; type checking; run-time environments; intermediate code generation; and compiler-construction tools.

162. Programming Languages

(4) HARDEKOPF, YU

Prerequisite: Computer Science 130A and Computer Science 138; open to computer science and computer engineering majors only.

Concepts of programming languages: scopes, parameter passing, storage management; control flow, exception handling; encapsulation and modularization mechanism; reusability through genericity and inheritance; type systems; programming paradigms (imperative, object-oriented, functional, and others). Emerging programming languages and their development infrastructures.

165A. Artificial Intelligence

(4) WANG, TX., YAN

Prerequisite: Computer Science 130A

Introduction to the field of artificial intelligence, which seeks to understand and build intelligent computational systems. Topics include intelligent agents, problem solving and heuristic search, knowledge representation and reasoning, uncertainty, probabilistic reasoning, and applications of AI.

165B. Machine Learning

(4) WANG, W., DING

Prerequisite: Computer Science 130A.

Covers the most important techniques of machine learning (ML) and includes discussions of: well-posed learning problems; artificial neural networks; concept learning and general to specific ordering; decision tree learning; genetic algorithms; Bayesian

learning; analytical learning; and others.

170. Operating Systems

(4) WOLSKI, GUPTA T.

Prerequisite: Computer Science 130A; and, Computer Science 154 or ECE 154 (may be taken concurrently); open to computer science, computer engineering or electrical engineering majors only.

Basic concepts of operating systems. The notion of a process; interprocess communication and synchronization; input-output, file systems, memory management.

171. Distributed Systems

(4) EL ABBADI

Prerequisite: Computer Science 130A.

Not open for credit to students who have completed ECE 151.

Distributed systems architecture, distributed programming, network of computers, message passing, remote procedure calls, group communication, naming and membership problems, asynchrony, logical time, consistency, fault-tolerance, and recovery.

174A. Fundamentals of Database Systems

(4) SU

Prerequisite: Computer Science 130A

Recommended Preparation: Students are strongly encouraged to complete Computer Science 56 prior to enrolling in Computer Science 174A

Database system architectures, relational data model, relational algebra, relational calculus, SQL, QBE, query processing, integrity constraints (key constraints, referential integrity), database design, ER and object-oriented data model, functional dependencies, lossless join and dependency preserving decompositions, Boyce-Codd and Third Normal Forms.

176A. Introduction to Computer Communication Networks

(4) ALMEROTH, BELDING

Prerequisite: CMPSC 32 with a grade of C or better; PSTAT 120A or ECE 139; open to computer science, electrical engineering, and computer engineering majors only.

Not open for credit to students who have completed Computer Science 176 or ECE 155 or ECE 155A.

Recommended preparation: PSTAT 120B.

Basic concepts in networking, the OSI model, error detection codes, flow control, routing, medium access control, and high-speed networks.

176B. Network Computing

(4) ALMEROTH

Prerequisite: Computer Science 176A.

Not open for credit to students who have completed ECE 155B or 194W.

Focus on networking and web technologies used in the Internet. The class covers socket programming and web-based techniques that are used to build distributed applications.

176C. Advanced Topics in Internet Computing

(4) GUPTA, A.

Prerequisite: Computer Science 176B.

General overview of wireless and mobile networking, multimedia, security multicast, quality of service, IPv6, and web caching. During the second half of the course, one or more of the above topics are studied in greater detail.

177. Computer Security

(4) KRUEGEL, VIGNA

Prerequisite: Computer Science 170 (may be taken concurrently).

Introduction to the basics of computer security and privacy. Analysis of technical difficulties of producing secure computer information systems that provide guaranteed controlled sharing. Examination and critique of current systems, methods, certification.

178. Introduction to Cryptography

(4) EGECIOGLU

Prerequisites: Computer Science 24 and Computer Science 40 with a grade of C or better; and PSTAT 120A or 121A or ECE 139 or permission of instructor.

An introduction to the basic concepts and techniques of cryptography and cryptanalysis. Topics

include: The Shannon Theory, classical systems, the Enigma machine, the data encryption standard, public key systems, digital signatures, file security.

180. Computer Graphics

(4) YAN, L.

Prerequisite: Computer Science 130A or consent of instructor.

Overview of OpenGL graphics standard, OpenGL state machine, other 3D graphics libraries, 3D graphics pipeline, 3D transformations and clipping, color model, shading model, shadow algorithms, texturing, curves and curved surfaces, graphics hardware, interaction devices and techniques.

181. Introduction to Computer Vision

(4) WANG, Y-F., TURK

Prerequisite: Upper-division standing.

Same course as ECE 181B.

Overview of computer vision problems and techniques for analyzing the content images and video. Topics include image formation, edge detection, image segmentation, pattern recognition, texture analysis, optical flow, stereo vision, shape representation and recovery techniques, issues in object recognition, and case studies of practical vision systems.

184. Mobile Application Development

(4) HOLLERER

Prerequisite: Computer Science 56 and Computer Science 130A.

An introduction to programming mobile computing devices. Students will learn about and study the shift in software development from desktop to mobile device applications. Topics will include software engineering and design practices, advances in programming practice, and support tools for mobile application development and testing. Students will develop and deploy mobile applications as part of their course work.

185. Human-Computer Interaction

(4) HOLLERER

Prerequisite: Upper-division standing in computer science, computer engineering, or electrical engineering majors.

Recommended preparation: Students are strongly encouraged to complete Computer Science 56 prior to enrolling in Computer Science 185. Proficiency in the Java/C++ programming language, some experience with user interface programming.

The study of human-computer interaction enables system architects to design useful, efficient, and enjoyable computer interfaces. This course teaches the theory, design guidelines, programming practices, and evaluation procedures behind effective human interaction with computers.

189A. Senior Computer Systems Project

(4) BULTAN, KRINTZ

Prerequisite: Computer Science 56; Senior standing in computer engineering, computer science, or electrical engineering; consent of instructor.

Not open for credit to students who have completed Computer Science 172 or ECE 189A.

Student groups design a significant computer-based project. Multiple groups may cooperate toward one large project. Each group works independently; interaction among groups is via interface specifications and informal meetings. Project for follow-up course may be different.

189B. Senior Computer Systems Project

(4) BULTAN, KRINTZ

Prerequisite: CMPSC 189A; Senior standing in computer engineering, computer science, or electrical engineering; consent of instructor.

Not open for credit to students who have completed ECE 189A or ECE 189B.

Student groups design a significant computer-based project. Multiple groups may cooperate toward one large project. Each group works independently; interaction among groups is via interface specifications and informal meetings. Project for course may be different from that in first course.

190AA-ZZ. Special Topics in Computer Science

(4) STAFF

Prerequisite: consent of instructor.

May be repeated with consent of the department chair.

Courses provide for the study of topics of current interest in computer science: A. Foundations; B. Software Systems; C. Programming languages and software engineering; D. Information management; E. Architecture; F. Networking; G. Security; H. Scientific computing; I. Intelligent and interactive systems; N. General

192. Projects in Computer Science

(1-5) STAFF

Prerequisite: consent of instructor.

Students must have a minimum 3.0 GPA.

May be repeated to a maximum of 8 units with consent of the department chair but only 4 units may be applied to the major.

Projects in computer science for advanced undergraduate students.

193. Internship in Industry

(1-4) STAFF

Prerequisites: consent of instructor and department chair.

Not more than 4 units per quarter; may not be used as a field elective and may not be applied to science electives. May be repeated with faculty/chair approval to a maximum of 4 units.

Special projects for selected students. Offered in conjunction with selected industrial and research firms under direct faculty supervision. Prior departmental approval required. Written proposal and final report required.

196. Undergraduate Research

(2-4) STAFF

Prerequisite: Students must: (1) have attained upper-division standing (2) have a minimum 3.0 grade-point average for preceding three quarters, (3) have consent of instructor.

May be repeated for up to 12 units. No more than 4 units may be applied to departmental electives.

Research opportunities for undergraduate students. Students will be expected to give regular oral presentations, actively participate in a weekly seminar, and prepare at least one written report on their research.

199. Independent Studies in Computer Science

(1-4) STAFF

Prerequisites: upper-division standing; must have completed at least two upper-division courses in computer science.

Must have a minimum 3.0 grade-point average for the preceding three quarters. May be repeated with consent of chair. Students are limited to 5 units per quarter and 30 units total in all 198/199 courses combined. May not be used for credit towards the major.

Independent study in computer science for advanced students.

GRADUATE COURSES

Graduate courses for this major can be found in the UCSB General Catalog.

Electrical & Computer Engineering

Department of Electrical and Computer Engineering, Building 380, Room 101;
Telephone (805) 893-2269 or (805) 893-3821
Web site: www.ece.ucsb.edu

Chair: Nadir Dagli
Vice Chair: Luke Theogarajan

Faculty

Rod C. Alferness, Ph.D., University of

Michigan, Distinguished Professor and Dean (integrated optoelectronics, optical switching technology and switched optical networks)

Mahnoosh Alizadeh, Ph.D., UC Davis, Assistant Professor (Smart power grids, demand response and renewable energy integration, cyber-physical systems, network control)

Kaustav Banerjee, Ph.D., UC Berkeley, Professor (high performance VLSI and mixed signal system-on-chip designs and their design automation methods; single electron transistors; 3D and optoelectronic integration)

Ilan Ben-Yaacov, Ph.D., UC Santa Barbara, Lecturer (semiconductor device physics and electronic devices, power electronics, engineering education)

Daniel J. Blumenthal, Ph.D., University of Colorado at Boulder, Professor (fiber-optic networks, wavelength and subcarrier division multiplexing, photonic packet switching, signal processing in semiconductor optical devices, wavelength conversion, microwave photonics)

John E. Bowers, Ph.D., Stanford University, Distinguished Professor (high-speed photonic and electronic devices and integrated circuits, fiber optic communication, semiconductors, laser physics and mode-locking phenomena, compound semiconductor materials and processing)

Forrest D. Brewer, Ph.D., University of Illinois at Urbana-Champaign, Professor (VLSI and computer system design automation, theory of design and design representations, symbolic techniques in high level synthesis)

James Buckwalter, Ph.D., California Institute of Technology, Professor (RF and mixed-signal CMOS integrated circuits, high-speed communications systems)

Katie A. Byl, Ph.D., Massachusetts Institute of Technology, Associate Professor (robotics, autonomous systems, dynamics, control, manipulation, locomotion, machine learning)

Shivkumar Chandrasekaran, Ph.D., Yale University, Professor (numerical

