long-term careers in research, technical development, and applications. Graduates of the B.S. and B.A. programs 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 issues.
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. UCSB students in majors other than computer science major can petition to the Department of Computer Science for consideration for admission via change-of-major once they complete the minimum requirements (specified on the departmental web pages) for doing so. Computer Science majors have priority when registering for all Computer Science courses.

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

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.

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.

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/change of college.

Students may petition to enter the Computer Science major when the following requirements are met:
1. A cumulative grade point average of at least 3.0;
2. Satisfactory completion of Computer Science 16 and 24 with a cumulative GPA of 3.2 or higher; First takes only
3. Satisfactory completion of Math 3A, 3B, 4A, 4B and Computer Science 40 with a cumulative GPA of 3.0 or higher; First takes only
4. An understanding of professional, ethical, and social responsibilities.
5. An ability to communicate effectively.
6. An ability to analyze the impact of computing on individuals, organizations, and society, including ethical, legal, security, and global policy issues.
7. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
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.

Computer Science Courses

LOWER DIVISION

1. Seminar on the Field of Computer Science
   (1) FRANKLIN
   Overview the potential of, and opportunities available from, the field of computer science. Topics include an overview of how computers work and the interesting ways in which computers can be applied to solve important and high-impact technological, social, and cutting-edge research problems.

8. Introduction to Computer Science
   (4) CONRAD, FRANKLIN
   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.

11AA-ZZ Programming Language Laboratory
   (1) FRANKLIN
   Different sections may be repeated. Sections not always offered.
   Recommended preparation: knowledge of at least one programming language.
   Self-paced course to allow a student who already possesses a working knowledge of at least one programming language an opportunity to learn other languages of interest.

16. Problem Solving with Computers I
   (4) CONRAD, KIRNITZ
   Prerequisite: Math 3A with a C or better (may be taken concurrently) CS 8, Engineering 3, 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 tools.

24. Problem Solving with Computers II
   (4) FRANKLIN, COSTANZO
   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 data structures, object-oriented design and development, algorithms for manipulating these data structures and their runtime analyses. Data structures introduced include stacks, queues, lists, trees, and sets.

32. Object Oriented Design and Implementation
   (4) HOLLIERER
   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 Math and statistics 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.
48. Computer Science Project
(4) CAPPELLO
Prerequisite: Computer Science 32 with a grade of C or better, and Computer Science 56 with a grade of C or better (can be taken concurrently).

Team-based project development. Topics include software 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 high level, 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) ZHENG, FRANKLIN
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-2Z. Undergraduate Seminar in Computer Science
(1-4) STAFF
Prerequisite: 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. 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) PETZOLD
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 informal 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) GONZALEZ
Prerequisites: Computer Science 40 and 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.

Concepts and principles of data structures such as: stacks, queues, linked lists, trees, recursion, and fundamental algorithms. Students learn and demonstrate their applications. Correctness proofs and techniques for the design of correct programs. Techniques of computer organization, operating systems, memory management. Design and creation of algorithms. Applications of techniques to problems from several disciplines. NP - completeness.

130B. Data Structures and Algorithms II
(4) GONZALEZ, 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) EGEKIOGLU
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 computational models. Introduction to Turing machines and computational complexity.

140. Parallel Scientific Computing
(4) 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, parallel program parallelization methodologies, parallel performance evaluation and optimization, parallel numerical algorithms and applications with different performance trade-offs.

153A. Hardware/Software Interface
(4) SHERWOOD, CHONG
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) SHERWOOD, CHONG
Prerequisite: Computer Science 32 and Computer Science 64.

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, and the interconnection of computer systems, software versus hardware trade-offs.

160. Translation of Programming Languages
(4) SHERWOOD
Prerequisite: Computer Science 64 or Electrical Engineering 154; Computer Science 130A; and, Computer Science 130B; open to computer science and computer engineering majors only.

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

162. Programming Languages
(4) HARDEKOPP, KRINTZ
Prerequisite: Computer Science 130A and Computer Science 138; open to computer science and computer engineering majors only.

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

165A. Artificial Intelligence
(4) TURK
Prerequisite: Computer Science 130A.

Introduction to the field of artificial intelligence, which seeks to understand 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) SINGH
Prerequisite: Computer Science 130A.

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

Advanced 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.

174. Fundamentals of Database Systems
(4) BU
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 models, relational algebra, relational calculus, SQL, ODE, 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) ALMERO, 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) ZHAO, VIGNA
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
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
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
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
Prerequisite: Computer Science 130A or consent of instructor.
Overview of OpenGL graphics standard, OpenGl state machines, 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.

181B. Introduction to Computer Vision
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.

185. Human-Computer Interaction
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
Prerequisite: 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 follow-up course may be different.

189B. Senior Computer Systems Project
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
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

Electrical & Computer Engineering
Department of Electrical and Computer Engineering, Building 380, Room 101; Phone (805) 893-2269 or (805) 893-3821
Web site: www.ece.ucsb.edu
Chair: Joao Hespanha
Vice Chair: B.S. Manjunath
Faculty
Rod C. Alferness, Ph.D., University of Michigan, Professor and Dean (integrated optoelectronics, optical switching technology and switched optical networks)
Kaustaw 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)
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, 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)

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