

Department of Computer Science.

The undergraduate major in computer science has a dual purpose: to prepare students for advanced studies and research and to provide training for a variety of careers in business, industry, and government.

Under the direction of the Associate Dean for Undergraduate Studies, academic advising services are jointly provided by advisors in the College of Engineering, as well as advisors in the department. A faculty advisor is also available to each undergraduate class for further academic program planning.

## Program Goals for Undergraduate Programs

The goal of the computer science undergraduate program is to prepare future generations of computer professionals for long-term careers in research, technical development, and applications. Graduates 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.

## 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

#### 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

(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 with a grade of C or better (may be taken concurrently).

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

Legal repeat of Computer Science 24A.

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).

#### 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 15B

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.

#### 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

#### 100. Introduction to Teaching Methods in Computer Science

(4) MIRZA

Prerequisite: consent of instructor.

May be repeated to a maximum of 12 units.

Designed to train outstanding undergraduates for learning assistant positions in CS courses. Lecture/discussion surveys current research and best practices in CS pedagogy including student development theories, different pedagogical techniques,

and methods for assessing learning. Students gain experience working one-on-one with students, fostering positive learning environments, and providing feedback on student work. Students who successfully complete this course will earn units by serving as an apprentice undergraduate learning assistant.

### 110. Introduction to Research in Computer Science

(4) MIRZA  
Prerequisite: Computer Science 40 and Computer Science 32; consent of instructor.

Defining a CS research problem, finding and reading technical papers, oral communication, technical writing, and independent learning. Course participants work in teams as they apprentice with a CS research group to propose an original research problem and write a research proposal.

### 111. Introduction to Computational Science

(4) GILBERT, MATNI  
Prerequisite: Mathematics 4B with a grade of C or better; Mathematics 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 the numerical algorithms that form the foundations of data science, machine learning, and computational science and engineering. Matrix computation, linear equation systems, eigenvalue and singular value decompositions, numerical optimization. The informed use of mathematical software environments and libraries, such as python/numpy/scipy.

### 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.

Data structures and applications with proofs of correctness and analysis. Hash tables, priority queues (heaps); balanced search trees. Graph traversal techniques and their applications. 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. 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 with a grade of C or better; Mathematics 6A 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.

### 148. Computer Science Project

(4) CONRAD  
Prerequisite: Computer Science 32 with a grade of C or better; open to Computer Science majors only.

Not open for credit to students who have completed Computer Science 48 with a grade of C or better.

CMPSC 148 is a legal repeat of CMPSC 48.

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 final projects.

### 153A. Hardware/Software Interface

(4) KRINTZ  
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) MATNI  
Prerequisite: Computer Science 32 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.

### 156. Advanced Applications Programming

(4) CONRAD  
Prerequisite: Computer Science 24 and 32 with a grade of C or better; open to computer science and computer engineering majors only.

Not open for credit to students who have completed Computer Science 56 with a grade of C or better.

CMPSC 156 is a legal repeat of CMPSC 56.

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, automatic memory management, and application development, management, and maintenance tools, third-party library use, version control, software testing, issue tracking, code review, and working with legacy code.

### 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, FENG  
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, YX., 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 and probabilistic reasoning, machine learning, reinforcement learning, and responsible AI.

### 165B. Machine Learning

(4) WANG, W., DING  
Prerequisite: Computer Science 130A[the 'recommended preparation' should be on the next line and not italicized]

Recommended preparation: Computer Science 111.

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.

### 172. Software Engineering

(4) BULTAN  
Prerequisite: Computer Science 130A; computer science or computer engineering majors only, or by consent of department.

Software engineering is concerned with long-term, large-scale programming projects. Software management, cost estimates, problem specification and analysis, system design techniques, system testing and performance evaluation, and system maintenance. Students will design, manage, and implement a medium-sized project.

### 174A. Fundamentals of Database Systems

(4) SU  
Prerequisite: Computer Science 130A  
Recommended Preparation: Students are strongly encouraged to complete Computer Science 56 or Computer Science 156 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.

### 174B. Design and Implementation Techniques of Database Systems

(4) SU, YAN  
Prerequisite: Computer Science 130B.  
Recommended Preparation: Students are strongly encouraged to complete Computer Science 56 or Computer Science 156 prior to enrolling in Computer Science 174B.

Q[insert the next line here to separate the course description from the 'recommended preparation']queries and processing, optimizer, cost models, execution plans, rewriting rules, access methods, spatial indexing; transactions, ACID properties, concurrency control, serializability, two-phase locking, timestamping, logging, checkpointing, transaction abort and commit, crash recovery; distributed databases.

### 176A. Introduction to Computer Communication Networks

(4) 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) GUPTA, A.

**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 176A

[the 'recommended preparation' should not be italicized]Recommended preparation: PSTAT 120B.

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) ANANTH

*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.

*Prerequisite:* Upper-division standing.

Same course as ECE 181.

Not open for credit to students who have completed ECE/CMPSC 181B with a grade of C or better. ECE/CMPSC 181 is a legal repeat of ECE/CMPSC 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 or Computer Science 156; 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 or Computer Science 156 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 48 or 56 or 148 or 156 or 172; Senior standing in computer science or computer engineering.

Not open for credit to students who have completed 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 science or computer engineering.

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.

### 196B. Undergraduate Research

(2-4) STAFF

*Prerequisite:* Students must (1) have taken 4 letter-graded units of CMPSC 196, (2) have consent of instructor.

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

Advanced research for undergraduate students, by petition after completing a minimum of 4 units of CMPSC 196 for a letter grade. The student will propose a specific research project and make a public presentation of final results. Evaluation and grade will be based on feedback from the research faculty advisor and one other faculty member.

### 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.