Computer Science

In the College of Sciences

OFFICE: Geology/Mathematics/Computer Science 413
TELEPHONE: (619) 594-6191

The B.S. degree in Computer Science is accredited by the Computer Science Accreditation Commission of the Computing Sciences Accreditation Board.

Faculty

Emeritus: Baase-Mayers, Marovac, Vinge
Chair: Beck
Professors: Anantha, Beck, Carroll, Donald, Stewart, Swiniarski, Tarokh, Vuskovic
Associate Professors: Eckberg, Valafar, Whitney
Assistant Professors: Lewis, Roch
Lecturers: Bajic, Riggins
Adjunct: Root

Offered by the Department

Master of Science degree in computer science.
Major in computer science with the B.S. degree in applied arts and sciences.
Minor in computer science.
Certificate in geographic information science.

The Major

Computer Science is the study of computers and their applications. It is concerned with methods for storing and retrieving information, with the design and use of languages for writing computer programs, with the hardware systems that interpret such languages, and with the theoretical principles that form the foundations of computing. Computer Science includes a wide variety of specialties and application areas such as artificial intelligence, robotics, graphics, systems programming, simulation, and computer networks.

The Bachelor of Science in Computer Science is designed to provide students with a fundamental understanding of modern computing methodology and programming practices along with a complementary knowledge of hardware. The first two years provide the basic preparation in programming, data structures and architecture. The final two years are devoted to more advanced fundamentals and specialized electives.

Computers are used to store and manage information, to analyze scientific data, and in a wide variety of other applications. Computing technology is found in an almost limitless number of settings, ranging from automobiles to household appliances to toys. Because of this, a wide range of jobs are open to people trained in Computer Science. Employment opportunities are expected to remain very strong.

Impacted Program

The computer science major is an impacted program. Students must enter the University under the computer science premajor code (07010). To be admitted to the computer science major, students must meet the following criteria:

a. Complete with a grade of C or higher: Computer Science 107, 108, 237; and Mathematics 150, 151, 245. These courses cannot be taken for credit/no credit (Cr/NC);
b. Complete lower division General Education requirements in written and oral communication and critical thinking in the English language;
c. Have a cumulative GPA of 2.75 or higher;
d. To gain entry into the major, students must fulfill the premajor requirements described in the catalog in effect at the time they declare the premajor at SDSU (assuming continuous enrollment).

To complete the major, students must fulfill the degree requirements for the major described in the catalog in effect at the time they are accepted into the premajor at SDSU (assuming continuous enrollment).

Computer Science Major

With the B.S. Degree in Applied Arts and Sciences
(Premajor Code: 07010) (Major Code: 07011)

All candidates for a degree in applied arts and sciences must complete the graduation requirements listed in the section of this catalog on "Graduation Requirements."

A minor is not required for this major.

Preparation for the Major. Computer Science 107, 108, 237; Mathematics 150, 151, 245, 254; Statistics 250; Physics 195, 195L, 196, 196L, or Chemistry 200, 201, or Biology 201A, 201B; and two additional science courses selected with approval of a computer science adviser. These must be courses for sciences or engineering majors or have a strong emphasis on quantitative methods. (41-43 units)

Upper Division Writing Requirement. Passing the University Writing Examination or completing one of the approved writing courses with a grade of C (2.0) or better.

Major. A minimum of 37 upper division units to include Computer Science 310, 320, 370, 440, 490, 530, 560, 570; at least one course selected from Mathematics 541, 579, Statistics 350A, 550, or 551A; and 12 units of computer science electives selected with the approval of a computer science major adviser. At least nine units of electives must be in computer science. The student must complete an outline for the major and file a copy signed by a major adviser with the Office of Advising and Evaluations.

Computer Science Minor

The minor in computer science consists of a minimum of 18-24 units in computer science and mathematics to include Computer Science 107, 108; and at least 12 upper division units, or at least nine upper division units if the student completes a full calculus sequence, i.e., Mathematics 121 and 122, or 150 and 151. The courses selected are subject to the approval of the minor adviser.

Students must officially declare the minor before taking any upper division computer science courses. To be admitted to the computer science minor, students must meet the following criteria:

a. Complete Computer Science 107 and 108 with a grade of C (2.0) or higher;
b. Complete lower division General Education requirements in written and oral communication and critical thinking in the English language;
c. Have a cumulative GPA of 2.75 or higher; and
d. Complete at least nine units in the department of the student's major, including at least six units of upper division courses.

Courses in the minor may not be counted toward the major, but may be used to satisfy preparation for the major and general education requirements, if applicable. A minimum of six upper division units must be completed in residence at San Diego State University.

Geographic Information Science Certificate*

The purpose of the program is to prepare students to acquire, manage, and visualize geospatial data in public and private organizations. Students must apply for admission to the program before the completion of 12 certificate units and must complete the required units with a 2.5 grade point average.

The certificate requires 27 units distributed between the Departments of Computer Science and Geography as follows: 12-15 units selected from Computer Science 107, 108, 220, 310, 320, 503, 514, 520, 535, 551, 575 and 12-15 units selected from Geography 381, 484, 488, 584, 585, 588. Courses with relevant content may be substituted for the computer science and geography courses with the approval of the certificate adviser. Courses in the certificate may be counted toward the major in computer science if applicable.

* Additional prerequisites required for this certificate.

Courses (CS)

LOWER DIVISION COURSES

101. Information Technology and Society (3)
Information technology for non-majors. Basic concepts and definitions needed for elementary understanding of computers, software, telecommunications and the Internet, information systems, and social impact of information technology.

105. Visual Basic Programming (3)
Programming and problem solving using Visual Basic programming language on the PC.

106. Introduction to Computer Programming with FORTRAN (3) I, II (CAN CSCI 4)
Prerequisite: Satisfaction of the Entry-Level Mathematics requirement.
Introduction to problem solving on a computer, design of algorithms, and use of FORTRAN language. Extensive programming.

107. Introduction to Computer Programming (3) I, II, S
Prerequisite: Satisfaction of the Entry-Level Mathematics requirement.
Programming methodology and problem solving. Basic concepts of computer systems, algorithm design and development, data types, program structures. Extensive programming in Java.

108. Intermediate Computer Programming (3) I, II, S
Prerequisites: Qualification on the Mathematics Departmental Placement Examination, Part IA; and Computer Science 107.
Further training in program design and development. Introduction to data structures: stacks, queues, linear lists, trees, sets, and recursion. Extensive programming in Java.

205. Introduction to Computational Programming and Visualization (3)
Prerequisite: First semester calculus (either Mathematics 120 or 121 or 150).
Problem solving skills for needs of science. Use of computing and software tools of computational science introduced to gain competence in computer communications, programming and visualization. Supervised computer laboratory.

220. UNIX and the C Programming Language (3) I, II
Prerequisite: Computer Science 108.
Introduction to the UNIX operating system: shell programming, major system services and utilities. The C language: its features and their significance in the UNIX programming environment.

237. Machine Organization and Assembly Language (3) I, II
Prerequisite: Computer Science 108.
General concepts of machine and assembly language, data representation, looping and addressing techniques, arrays, subroutines, macros. Extensive assembly language programming.

296. Experimental Topics (1-4)
Selected topics. May be repeated with new content. See Class Schedule for specific content. Limit of nine units of any combination of 296, 496, 596 courses applicable to a bachelor's degree.

299. Special Study (1-3)
Prerequisite: Consent of instructor.
Individual study. Maximum credit six units.

UPPER DIVISION COURSES (Intended for Undergraduates)

310. Data Structures (3) I, II, S
Prerequisites: Computer Science 108 and Mathematics 245.
Representations and operations on basic data structures. Arrays, linked lists, stacks, queues, and recursion; binary search trees and balanced trees; hash tables, dynamic storage management; introduction to graphs. An object oriented programming language will be used.

320. Programming Languages (3) I, II
Prerequisite: Computer Science 108.
Principles of high-level programming languages, including formal techniques for syntax specification and implementation issues. Languages studied should include at least C++, FORTRAN, and LISP.

370. Computer Architecture (3) I, II
Prerequisite: Computer Science 237.
Logic gates, combinational circuits, sequential circuits, memory and bus system, control unit, CPU, exception processing, traps and interrupts, input-output and communication, reduced instruction set computers, use of simulators for analysis and design of computer circuits, and traps/interrupts.

425. Tcl and Tk Interface Programming (3)
Prerequisite: Computer Science 220.
Presentation of Toolkit Command Language (Tcl) and Toolkit (Tk) languages, a portable programming environment for creating graphical user interfaces under X Windows, Microsoft Windows, and Macintosh. Writing scripts for Tcl, Tk, and extensions such as Expect.

435. Advanced Java Programming (3)
Prerequisite: Computer Science 310.
Object oriented modeling techniques and tools; use cases; UML models and diagrams. Exception handling, I/O with objects, random access I/O, subclasses and inheritance, overloading versus overriding, interfaces and abstract classes, threads, cloning, packages, documentation aids, archiving and compression, iterators and comparators.

470. UNIX System Administration (3)
Prerequisite: Computer Science 220.
Installing the UNIX operating system on a UNIX workstation, adding user accounts, backing up and restoring user files, installing windows, adding network capabilities, adding printers and other peripherals.

490. Senior Seminar (1)
Prerequisite: Fifteen units of upper division computer science courses.
Preparation and delivery of oral presentations on advanced topics in computer science. General principles of organization and style appropriate for presenting such material.

496. Experimental Topics (1-4)
Selected topics. May be repeated with new content. See Class Schedule for specific content. Limit of nine units of any combination of 296, 496, 596 courses applicable to a bachelor's degree.

498. Directed Readings in Computer Science Literature (1)
Prerequisite: Credit or concurrent registration in the upper division computer science course in which readings are to be undertaken.
Individually directed readings in computer science literature. May be repeated for a maximum of three units, taken each time from a different instructor.

499. Special Study (1-3) I, II
Prerequisite: Consent of instructor.
Individual study. Maximum credit six units.

UPPER DIVISION COURSES
(Also Acceptable for Advanced Degrees)

501. Computational Software (3)
Prerequisites: Computer Science 106 and 310.
Design and implementation of software for computational science. Makefiles in UNIX environment, efficient Fortran and 00 programming, use of common application libraries, file and source code management, software documentation, construction of libraries and applications. Designed for computational science students. Computer science majors must obtain adviser approval.

503. Scientific Database Techniques (3)
Prerequisites: Computer Science 205, 310, and Mathematics 245.
Fundamental data models for handling scientific data, including flat file, indexed compressed files, relational databases, and object oriented databases, and their associated query technologies; e.g. file formats, input/output libraries, string searching, structured query language, object-oriented structured query language, hypertext markup language/common gateway interface, and other specialized interfaces. Designed for computational science students. Computer science majors must obtain adviser approval. See Computer Science 514.

505. Parallel Computing (3)
Prerequisite: Computer Science 310. Recommended: Computer Science 501.
Motivations and methods of high performance computing. Modern computer architecture characteristics, uniprocessor programming and tuning, shared and distributed memory programming techniques, benchmarking. Designed for computational science students. Computer science majors must obtain consent of adviser.

514. Database Theory and Implementation (3)
Prerequisites: Computer Science 310 and Mathematics 245.
Database systems architecture. Storage structures and access techniques. Relational model, relational algebra and calculus, normalization of relations, hierarchical and network models. Current database systems.

520. Advanced Programming Languages (3)
Prerequisites: Computer Science 237, 310, and 320.
Object oriented programming, concurrent programming, logic programming. Implementation issues.

524. Compiler Construction (3)
Prerequisites: Computer Science 237, 310, and 320.
Syntactical specification of languages. Scanners and parsers. Precedence grammars. Run-time storage organization. Code generation and optimization.

530. Systems Programming (3) I, II
Prerequisites: Computer Science 237 and 310.
Design and implementation of system software. Relationship between software design and machine architecture. Topics from assemblers, loaders and linkers, macro processors, compilers, debuggers, editors. Introduction to software engineering and review of programming fundamentals and object oriented concepts. Large project in object oriented programming is required. Not acceptable for the M.S. degree in computer science.

532. Software Engineering (3)
Prerequisites: Computer Science 320 and 530.
Theory and methodology of programming complex computer software. Analysis, design, and implementation of programs. Team projects required.

535. Object-Oriented Programming and Design (3)
Prerequisites: Computer Science 310 and 320.
Basic concepts of object-oriented programming; classes, objects, messages, data abstraction, inheritance, encapsulation. Object-oriented design methodology.

550. Artificial Intelligence (3)
Prerequisites: Computer Science 108 and either Mathematics 245 or 523.
Heuristic approaches to problem solving. Systematic methods of search of the problem state space. Theorem proving by machine. Resolution principle and its applications.

551. User Interface Environments (3)
Prerequisites: Computer Science 310 and 320.
Design of user-machine interfaces in interactive systems. Problems faced by user of an interactive system; basic issues and principles involved in design and implementation of good and friendly user-machine graphical interfaces.

552. Artificial Intelligence II (3) II
Prerequisite: Computer Science 550.
Limitations of symbol-based approach to artificial intelligence from Computer Science 550. Presented alternatives are genetic and probabilistic approaches, connectionist and emergent representation and learning, natural language processing, intelligence measures and cognitive models. Seminal publications shaping these techniques.

553. Neural Networks (3)
Prerequisites: Computer Science 320 and Mathematics 254.
Principles of neural networks, their theory and applications.

555. Raster Computer Graphics (3)
Prerequisite: Computer Science 551.
Bit map graphics, algorithms to connect between different formats and enhancement of pictures.

556. Robotics: Mathematics, Programming, and Control (3)
Prerequisites: Computer Science 320, Mathematics 254, knowledge of the C programming language.
Robotic systems including manipulators, actuators, sensors, and controllers. Algebraic methods for spatial description of solid objects, manipulator kinematics and control. Robot programming languages and robot programming systems.

557. Computer Control Systems (3)
Prerequisites: Computer Science 310 and Mathematics 254.
Analysis and programming of real-time computer control systems, implementation of digital controllers including programming, intelligent control systems and fuzzy control.

558. Computer Simulation (3)
Prerequisites: Computer Science 310 and Statistics 550.
Methodology of simulation for discrete and continuous dynamic systems. State-of-the-art programming techniques and languages. Statistical aspects of simulation. Students will design, program, execute, and document a simulation of their choice.

559. Computer Vision (3)
Prerequisites: Computer Science 310 and Mathematics 254.
Algorithms and computer methods for processing of images. Visual perception as a computational problem, image formation, characterization of images, feature extraction, regional and edge detection, computer architectures for machine vision.

560. Algorithms and Their Analysis (3) I, II
Prerequisite: Computer Science 310.
Algorithms for solving frequently occurring problems. Analysis techniques and solutions to recurrence relations. Searching and sorting algorithms. Graph problems (shortest paths, minimal spanning trees, graph search, etc.). NP complete problems. Not acceptable for the M.S. degree in Computer Science.

561. Multimedia Systems (3)
Prerequisite: Computer Science 551.
System aspects of multimedia authoring, browsing, and database subsystem; digital representation for different media; audio and video; operating system support for continuous media applications; architectures; design and implementation of multimedia support systems; use of multimedia technology in software engineering.

562. Automata Theory (3)
Prerequisite: Mathematics 245 or 521A.
Definition of finite automata. Classification of finite automaton definable languages. Minimization of finite automata. Nondeterministic finite automata. Sequential machines with output. Regular sets and expressions. Introduction to grammars.

564. Introduction to Computability (3)
Prerequisite: Mathematics 245 or 523.
Definition of algorithm by abstract (Turing) machines. Universal Turing machines. Primitive recursive and recursive functions. The equivalence of the computational power of Turing machines and recursive functions. Limitations and capabilities of computing machines; the halting problem.

566. Queuing Theory (3)
Prerequisites: Computer Science 108 and Statistics 550.
Performance prediction of computer networks and other systems (e.g., inventory control, customer service lines) via queuing theory techniques. Operational analysis.

570. Operating Systems (3) I, II
Prerequisites: Computer Science 310, 370, and knowledge of the C programming language.
File systems, processes, CPU scheduling, concurrent programming, memory management, protection. Relationship between the operating system and underlying architecture. Not acceptable for the M.S. degree in Computer Science.

571. UNIX Network Administration (3)
Prerequisite: Computer Science 470.
Network administration for UNIX workstations and servers. File server, mail server, boot sever, and Web server. TC/IP administration, routing, subnetting, and NIS plus data base.

572. Microprocessor Architecture (3)
Prerequisites: Computer Science 370 and knowledge of the C programming language.
Architecture of state-of-the-art microprocessor. Internal pipeline, internal cache, external cache, and memory management. Programming a uniprocessor. Communication among computers in a distributed environment. Architecture and programming of a multiprocessor system.

574. Computer Security (3)
Prerequisites: Computer Science 310; Mathematics 245; Statistics 550; and credit or concurrent registration in Computer Science 570.
Principles of computer security and application of principles to operating systems, database systems, and computer networks. Topics include encryption techniques, access controls, and information flow controls.

575. Supercomputing for the Sciences (3)
Prerequisite: Extensive programming background in Fortran or C.
Interdisciplinary course, intended for all science and engineering majors. Advanced computing techniques developed for supercomputers. Overview of architecture, software tools, scientific computing and communications. Hands-on experience with CRAY.

576. Computer Networks and Distributed Systems (3)
Prerequisite: Credit or concurrent registration in Computer Science 570.
Local area networks and wide area networks; mechanisms for interprocess communication; rules for distribution of data and program functions.

578. ATM Networking (3)
Prerequisite: Computer Science 576.
Asynchronous transfer mode communication networks, including fundamental concepts, technologies, architectures, infrastructures, and interoperability of legacy technologies. Review of basic communication and networking concepts, including transmission media, multiplexing, link control protocols, and wide area networks.

580. Client-Server Programming (3)
Prerequisites: Computer Science 570 and knowledge of an object-oriented programming language. Recommended: Computer Science 576.
Client-server model, networking protocols for client-server programs, algorithmic issues in client-server programs, client-server protocols, implementing client-server applications.

581. Computational Linguistics (3)
(Same course as Linguistics 581.)
Prerequisites: Computer Science 320 or Linguistics 571; Linguistics 570 or Mathematics 245.
Basic concepts in computational linguistics including regular expressions, finite-state automata, finite-state transducers, weighted finite-state automata, and n-gram language models. Applications to phonology, orthography, morphology, syntax. Probabilistic models. Statistical techniques for speech recognition.

596. Advanced Topics in Computer Science (1-4) I, II
Prerequisite: Consent of instructor.
Selected topics in computer science. May be repeated with the approval of the instructor. See Class Schedule for specific content. Limit of nine units of any combination of 296, 496, 596 courses applicable to a bachelor's degree. Maximum credit of six units of 596 applicable to a bachelor's degree. Maximum combined credit of six units of 596 and 696 applicable to a 30-unit master's degree.

GRADUATE COURSES
Refer to Bulletin of the Graduate Division.

For additional courses useful to computer scientists, see:

Mathematics 541.	Introduction to Numerical Analysis and Computing
Mathematics 542.	Introduction to Numerical Solutions of Differential Equations
Mathematics 561.	Applied Graph Theory
Mathematics 579.	Combinatorics