Introduction

The Department of Computer Science is a new intercollege department, affiliated with both the College of Arts and Sciences and the College of Engineering. The department evolved from the Computer Science Group, a graduate interdisciplinary group that has offered M.S. and Ph.D. degrees since 1967. Since 1973, the department has offered a program of study leading to the degree of Bachelor of Science. Students in Computer Science have the option of being in either the College of Arts and Sciences or the College of Engineering.

Computer science is the study of information and algorithms within the context of real and abstract computing devices. Computer scientists are interested in: the representation and storage of information; algorithms to access, display, edit, and transform information; programming and mathematical languages to express algorithms; and hardware and software processors to execute algorithms. These concerns lead both to theoretical investigations of computers, algorithms, and data and to practical developments in computer technology and applications.

The department currently offers undergraduate courses in programming, discrete structures, data structures, machine organization, computer design, operating systems, programming languages, compilers, artificial intelligence, design and analysis of algorithms, and theory of computation. The department also requires its majors to take courses in calculus, linear algebra, differential equations, physics, electronics, and numerical analysis. Slightly different distribution requirements in Humanities and Social Sciences are required for students in the two colleges.

The objective of Computer Science education is to develop professionally competent and broadly educated computer scientists. Undergraduate education is designed to prepare students for professional careers or graduate study; especially important is a foundation that will not become obsolete as technology advances and changes.

The computer field has a broad base of industrial and governmental users, providing many of the jobs suitable for the B.S. graduate. Typical jobs are systems analyst, systems programmer, scientific programmer, analyst, technical salesperson, hardware or software specialist. Above this base is a pyramid of producers and developers of computer systems, as well as teachers and researchers. Graduate education is appropriate for many of the jobs at these higher levels.

Walter L. Ruzzo, Assistant Professor of Computer Science (1977). B.S., 1968, California Institute of Technology; Ph.D., 1977, California (Berkeley). Computational complexity, operating systems theory.


Adjunct, Research, and Lecturer Appointments

Donald L. Adolphson, Assistant Professor of Quantitative Methods. Mathematical programming, stochastic models in operations research, computers in business.

Robert Gillespie, Lecturer of Computer Science; Assistant Provost for Computing. Operating systems, software engineering.

Alistair C. Holden, Professor of Electrical Engineering. Artificial Intelligence, pattern recognition of pictures and speech, symbol manipulation, list processing languages.

David L. Johnson, Professor of Electrical Engineering. Logical design of digital circuits, artificial intelligence, machine-learning modeling.

Victor Klee, Professor of Mathematics. Linear programming, network optimization, combinatorics, convexity, functional analysis.

Ralph T. Rockafellar, Professor of Mathematics. Linear programming, convex programming, optimization in directed graphs.

John S. Sobolewski, Research Assistant Professor of Computer Science and Medicine; Director, John L. Locke, Jr., Computer. Computer organization, data communication, medical applications, information storage and retrieval.
The Undergraduate Program

The Computer Science Department is associated with the College of Arts and Sciences and the College of Engineering. A student may be in either college, but, in each college, if the student successfully completes the Computer Science program, he or she will receive the degree of Bachelor of Science. A student typically enters the program during the sophomore year or at the beginning of the junior year.

Admission into the Computer Science Undergraduate Program requires the completion of a departmental application. Details of the admission policy follow the description of the program.

Requirements for a Degree of Bachelor of Science

There are four components:

(1) General Education Component (80 credits)
   (a) For a student in the College of Arts and Sciences:
       Humanities from the College distribution list: 20 credits
       Social Sciences from the College distribution list: 20 credits
       Free Electives: 40 credits
   (b) For a student in the College of Engineering:
       Humanities and Social Sciences: 30 credits with at least 10 in each
       Functional Techniques: 10 credits
       Free Electives: 40 credits

(2) Preparatory Component (45 credits)
    Math 124-5-6, 203, 238
    Physics 121-2-3, ECE 384, 411
    10 credits of Natural Science, Business, or Engineering (Math 391, 392, and Physics 384 are recommended)

(3) Computer Science Core Component (40 credits)
    CSci 201, 241, 321, 326, 378, 431, 441, 470, Math 464, and 6 credits of CSci 498

(4) Computer Science Elective Component (14 credits)
    A student may satisfy this requirement by taking additional courses on the Approved Computer Science Electives List or by taking graduate courses in Computer Science.

We give two sample curricula, the first for a student who starts a Computer Science major during the sophomore year and the second for a student who starts a Computer Science major at the beginning of the junior year. The number in parentheses indicates the number of credits; G, P, and C denote courses from the general education, preparatory, and Computer Science elective components respectively.

Four year program:

Freshman Year
   Spring: Math 126 (5), Phys 122 (4), G/P (6).

Sophomore Year
   Autumn: Math 205 (3), Phys 123 (4), CSci 241 (3), G/P (5).
   Winter: Math 238 (3), Phys 491 (3), G/P (9).
   Spring: G/P (13). 384

Junior Year
   Spring: CSci 326 (5), G/P/C (10).

Senior Year
   Spring: CSci 431 (3), CSci 498 (3), G/C (9).

Two Year Program:

Freshman and Sophomore years (90 credits)
   Math 124-5-6, 203, 238, Physics 121-2-3, CSci 201, and general education and preparatory component credits.

Junior Year
   Spring: CSci 326 (5), G/P/C (10).

Senior Year
   Spring: CSci 431 (3), CSci 498 (3), G/C (9).

It is recommended that students augment their computer science major with concentrated work in some application areas like Physics, Mathematics, Economics, Accounting, or Electrical Engineering. The reason for this recommendation is that, although computers are worth studying in themselves, they are also used as tools by other disciplines. Knowledge of another discipline will allow a Computer Scientist to apply his expertise in computers to that discipline. For students planning to go to graduate school in Computer Science, it is suggested that they take additional courses in mathematics, emphasizing senior level courses in abstract algebra, analysis, logic, or probability and statistics.

Admission Policy

1. Objectives and Justification for an Admission Policy

The objective of Computer Science education is to develop professionally competent and broadly educated computer scientists. Undergraduate education is designed to prepare students for professional
careers or graduate studies. Enrollment will be limited due to space, faculty, and budgetary limitations. Those applicants who, in the judgment of the faculty, are most likely to succeed in the program and contribute to the field will be admitted. Applications from minorities and women are encouraged.

2. Pre-Selection Requirements

A student may apply to enter the Computer Science program provided:

a. the following courses or equivalent have been completed:

Math 124, 125;
Physics 121;
Computer Science 201.

b. the student has completed at least 30 credit hours applicable to University graduation requirements.

c. the student has an overall University grade point average of at least 2.5.

d. the student has been admitted to the University or has completed an application to the University through the Office of Admissions.

3. How and When to Apply

Application to the Computer Science Undergraduate Program requires the completion of a departmental application form. Any student meeting the preselection requirements (Section 2 above) may apply. Applications may be obtained from the Computer Science Department.

Completed applications must be received by the department by: April 15 for Autumn Quarter; October 15 for Winter Quarter; January 15 for Spring Quarter; April 1 for Summer Quarter. TRANSCRIPTS ARE ALSO NEEDED BY THE ABOVE DATES. UW STUDENTS MUST ARRANGE FOR THEIR LATEST TRANSCRIPTS TO BE SENT FROM ADMISSIONS AND RECORDS.

4. Selection Process

The Computer Science Undergraduate Admissions Committee will meet each quarter to select students for admission for the coming quarter.

Selection of applicants will be made primarily on the basis of scholastic achievement and potential. In order to meet the objectives stated in Section 1 above, other selection criteria, such as relevant work experience and grades in Computer Science preparatory courses or in other mathematics, science, and engineering courses, may also be considered. Students who are denied admission may request reconsideration of their application; further information on this procedure can be obtained from the chairman of the admissions committee.

Undergraduate Courses

CSci 201 Introduction to Computer Science (3) W
A rigorous introduction to the theoretical and practical components of computer science: algorithms, programs, data structures, machines, computability, applications, social aspects. Prerequisite: MATH 124.

CSci 241 Programming (3) A
Basic algorithms, programming techniques, and basic concepts of a sophisticated high-level language. Prerequisite: 201 or permission.

CSci 321 Discrete Structures (3) A
Fundamentals of set theory, graph theory, Boolean algebra, and algebraic structures with applications in computing. Prerequisite: MATH 126.

CSci 326 Data Structures (3) Sp
Sequential and linked allocation of linear structures; tables, stacks, queues; in-core searching and sorting; circular and doubly linked lists; trees and threaded lists; dynamic memory allocation and garbage collection. Prerequisites: 321 and 378.

CSci 378 Machine Organization and Assembly Language (5) W
Differences and similarities in machine organization; central processors; fundamentals of machine language and addressing; assembly language programming, including macros; operating system interfaces. Prerequisite: 241

CSci 401 Introduction to Assemblers and Compilers (3) W
Fundamentals of assemblers, compilers, and interpreters; symbol tables; macro-processing; lexical analysis, syntactic analysis, semantic analysis and code generation for general purpose programming languages. listed jointly with EE 401. Prerequisite: EE 371 or CSci 378 or CSci/EE 478.

CSci 421 Introduction to the Analysis of Algorithms (3) A
Analysis of behavior of algorithms. Techniques for design of efficient algorithms. Methods for showing lower bounds on computational complexity. Discussion of particular algorithms for sorting, searching, set-manipulation, arithmetic, graph problems, pattern-matching, and their implementations. Prerequisite: 326.

CSci 431 Introduction to Theory of Computation (3) Sp
Fundamentals of automata theory and formal language theory; models of computation; Turing machines; space and time complexity; the halting problem. Prerequisites: 241 and 321.

CSci 441 Programming Languages (5) A
Designed to make the student reasonably fluent in several radically different languages, such as LISP, SNOBOL, APL, ALGOL 60, PASCAL, SIMULA 67, and others. Prerequisite: 378.
CSci 451 Introduction to Operating Systems (3) W
Principles of multiprogramming systems. Process management, resource management, and file systems. Prerequisite: 326 or permission.

CSci 470 Computer Design (4) AW
Fundamental gating circuits are developed into large logic gating structures. The use of these structures in the design of central processing units, memories, and peripheral equipment is illustrated. For computer science majors. Prerequisite: 376 or permission.

CSci 473 Introduction to Artificial Intelligence (3) W
This course introduces the principal ideas and developments in Artificial Intelligence. An introduction to LISP and SNOBOL is given as the basis of precise descriptions of AI processes. Theorem proving, problem-solving methods are introduced. The representation of knowledge in procedures, and in frames, and recently developed methods which utilize these ideas are discussed. Natural language analysis and synthesis are studied, involving conceptual representations, inference, and generation from conceptual representations. Prerequisite: CSci 326 and 441.

CSci 478 Computer Organization and Machine Language Programming (4) ASP
Differences and similarities in computer structure. Number representations, instruction codes, addressing techniques, subroutine and macro linkage and expansion, principles of assemblers, data structures (arrays, tables, lists), searching and sorting, input-output operations. Prerequisites: ENGR 161 and 346, or equivalent.

CSci 498 Senior Project (1-6, max. 6) AWSp
Consists of a report (and perhaps demonstration) describing a development, survey, or small research project completed by the student in an area of specialization. Objectives include: (1) applying and integrating the classroom material from several courses; (2) introducing the professional literature; (3) gaining experience in writing a technical document, and (4) enhancing employability through the evidence of independent work. The project may cover an area in computer science or an application to another field. Prerequisite: senior standing.

Graduate Courses

CSci 500 Computers and Society (2) W
Study of the impact of computer technology on present and future society; computer technology and economics; political, economic, cultural, social, and moral issues. Seminar with frequent guest lecturers and discussion leaders. Each student is required to complete a term project. Offered on credit/no credit basis only. Prerequisite: graduate standing in computer science or permission.

CSci 501 Compiler Construction I (3) Sp

CSci 502 Compiler Construction II (3) A
Translator-writing systems, incremental compilation. Design of production compilers. Prerequisite: CSci 501. (Offered alternate years.)

CSci 505 Concepts of Programming Languages (3) Sp
Basic concepts in programming languages, data structures (arrays, records, lists); program patterns, environments, control, evaluation, application, matching; relation to high-level machines. Prerequisites: 478 and working knowledge of at least one ALGOL-like programming language and one LISP-like programming language.

CSci 506 Formal Semantics (3) W
The course will cover some of the basic formalisms in semantics including flowchart schema, recursive schema, fixed-point semantics and the associated induction rules, lambda calculus and other interpretive models. Formal semantics will be viewed as providing foundations for formal definitions of programming languages, program interpretation, compiler verification, theory of program optimization and other meaning-preserving program transformations.

CSci 508 Representation and Handling of Data Structures (3) A
Linear lists (stacks, queues, deques); sequential and linked allocation; circular and doubly linked lists; trees, binary trees, and threaded trees; traversal algorithms; analysis of flow charts; path length of trees; garbage collection; dynamic storage allocation; data management on external media. Prerequisite: 478 or permission.

CSci 510 List Processing and String Manipulation (3) Sp
Structure of information sets and processes that reflect syntactic and semantic relationships. The generation and processing of structures such as lists and trees. Symbolic pattern recognition and manipulation. Concepts and applications of recent versions of languages such as LISP, SNOBOL, and FORMULA-ALGOL. Recent developments in languages for artificial intelligence.

CSci 518 Digital Signal Processing (4) Sp
Signal classification and representation; data collection, processing, and display. Emphasis on the derivation, evaluation, and application of various information-extraction algorithms and their realization on a digital computer. Offered jointly with EE 518. Prerequisite: graduate standing or permission.

CSci 520 Computer Science Seminar (1, max 9) AWSp
Weekly discussion by students and faculty or visitors on current topics of interest. Offered on credit/no credit basis only.

CSci 521 Design and Analysis of Algorithms (3) W
Models of computation for measuring the efficiency of algorithms. Principles of design of efficient algorithms: recursion, divide and conquer, balancing, dynamic programming, data structure selection, analysis of algorithms. Examples drawn from problems in sorting, searching, set manipulation, pattern matching, graphs, matrices, polynomials, and integers. Prerequisite: 508.
CSCI 522 Design and Analysis of Algorithms II (3) Sp
Analysis of algorithms more sophisticated than those treated in
CSCI 521. Design of efficient algorithms for special computing
environments such as logical networks and formulas, Turing machines,
list-processing machines, and on-line computation. Techniques for
proving lower bounds on complexity. Prerequisite: CSCI 521.

CSCI 531 Syntactic Analysis (3) A
Regular sets and finite automata. Context free grammars and
pushdown store automata. General parsing methods: top down and bottom
up. Restricted parsing methods: LR(k), LK(k), precedence grammars,
shift reduce parsing.

CSCI 532 Models of Computation (3) W
Models of computation and their equivalence. The halting problem
for Turing machines. Space and time complexity. Deterministic and
non-deterministic computation. Problems complete in non-deterministic
polynomial time and polynomial space. Petri nets and vector addition
systems.

CSCI 517, 538 Computability Theory (3, 3) ASp
Introduction to the theory of effective computability. Formulation
of the concepts of recursive and partial recursive function and
recursively enumerable set. Study of the relationship between a program
and the properties of the function computed by that program.
Introduction to reducibility, procedures, construction of undecidable
problems, degrees of unsolvability, recursive invariance, and
isomorphism. Classification of unsolvable problems, the hierarchical
hierarchy, the relation of the degree of unsolvability of a set to the
logical complexity required to describe the set. Connection with
mathematical logic, the theorems of Godel, Church, and Tarski. As time
permits, introduction of various related topics in computational
complexity. Prerequisite: MATH 502 or permission.

CSCI 540 Discrete System Simulation (3) A
Principles of simulation of discrete, event-oriented systems.
Model construction, simulation and validation, and relationship to other
techniques for system analysis and design. Use of special-purpose
simulation languages such as SIMULA and study of functional components
and data structures. Prerequisite: programming experience with ALGOL.

CSCI 541 Computer Measurements and Evaluation Techniques (3) W
Viewpoints, problems, and techniques in assessment of computer
systems and subsystems. Selection of models, analysis, simulation, and
instrumentation, with problem assignments making use of computers
available on campus.

CSCI 542 Central Processor Architecture (3) Sp
Several central processing units are examined at the gate level.
Included are the logic structures of: I/O bus, memory bus, ALU, address
modification, control logic, combinatorial and multiphase instructions,
access priority, cycle stealing, etc. Prerequisite: 470

CSCI 548 Computer Systems Architecture (3) W
Notations for describing computer systems. Powerful CPUs. Memory
organization. Channels and I/O processors. Micro programming. Stack
computers. Array and pipeline processors. Prerequisite: 478;
corequisite: 470 or permission.

CSCI 551 Operating Systems (3) Sp
Operating systems design and construction techniques. Systems
programming languages, concurrent programming, design methodologies,
protection, deadlock problems, virtual memory allocation, and other
topics. Study of the structure of different kinds of operating
systems. Prerequisite: 451 or permission.

CSCI 557 Computer Graphics (3) A
Generation and interpretation of pictures by computer with or
without human interaction. Graphics hardware, Display programming.
Picture transformations, Representations of pictures and their
attributes. Hidden line and surface problems. Graphics programming
languages and systems. Linguistic methods in picture analysis and
generation. Each student is required to complete a project on the
interactive graphics facility in the Computer Science Teaching
Laboratory. Prerequisite: 508.

CSCI 561 Computer Communications and Networks (3) A
Fundamentals of data transmission: coding, message formats, and
protocols; Job and data management problems; organization of computer
networks. A number of networks are studied, and students are expected
to prepare a class presentation of a network. Offered on credit/no
credit basis only. Prerequisite: graduate standing. (Offered
alternate years.)

CSCI 573 Artificial Intelligence I (3) A
Introduction to the use of the computer in non-numerical problem
solving. Survey of theorem proving, program manipulating, pattern
recognition, and inductive problem-solving techniques. Computer models
of human thought. Prerequisite: 476.

CSCI 574 Artificial Intelligence II (3) W
Continuation of studies of artificial intelligence systems,
emphasizing theorem proving, symbolic problem solving, pattern
recognition, and natural language data processing. Students are
required to do projects. Prerequisite: 573 or permission.

CSCI 590 Special Topics in Computer Science (*) ASp
Lectures and discussions of topics of current interest in computer
science. May not be offered every quarter; content may vary from one
offering to another. Prerequisite: permission.

CSCI 600 Independent Study or Research (*) ASpS
Offered on credit/no credit basis only.

CSCI 700 Master's Thesis (*) ASpS
Offered on credit/no credit basis only.

CSCI 800 Doctoral Dissertation (*) ASpS
Offered on credit/no credit basis only.
Approved Computer Science Electives

CSci 401 Introduction to Assemblers and Compilers (3)
CSci 421 Introduction to Analysis of Algorithms (3)
CSci 451 Introduction to Operating Systems (3)
CSci 473 Introduction to Artificial Intelligence (3)

The following courses offered by other departments are acceptable as Computer Science Electives.

Accounting 440 Accounting Systems (3)
Electrical Engineering 418 Introductory Communication Theory (3)
Electrical Engineering 475 Digital Systems (4)
Electrical Engineering 476 Logical Design of Digital Devices (3)
Electrical Engineering 479 Microcomputer System Design (4)
Geography 365 Computer Cartography (5)
Mathematics 391 Elementary Probability (3) *see Note 1
Mathematics 392 Elements of Statistics (3) *see Note 1
Mathematics 403 Introduction to Metamathematics (3)
Mathematics 407, 408 Mathematical Optimization Theory (3,3)
Mathematics 465 Numerical Analysis II (3)
Mathematics 466 Numerical Analysis III (3)
Physics 434 Application of Computers to Physical Measurement (3)
Psychology 475 Computing in Behavioral Sciences (3)
Quantitative Methods 404 Computer Programming for Business (4)
Quantitative Science 381 Introduction to Probability and Statistics (3) *see Note 1

*Note 1: Either Quantitative Science 381 or Mathematics 391 and 392 may be used as Computer Science Electives.