UNDERGRADUATE HANDBOOK FOR COMPUTER SCIENCE

Department of Computer Science
University of Washington
114 Sieg Hall, FR-35
Seattle, Washington 98195
(206) 543-1695
Introduction

The Department of Computer Science evolved from the Computer Science Group, a graduate interdisciplinary group that has offered M.S. and Ph.D. degrees since 1967. Since 1975, the department has offered an undergraduate program in which students can pursue a Bachelor of Science degree under 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.

The Faculty

The year the faculty member joined the University is in parentheses.


David B. Dekker, Associate Professor of Mathematics and Computer Science (1943), Graduate Program Advisor for Computer Science. B.S., 1941, California (Berkeley); M.S., 1943, Illinois Institute of Technology; Ph.D., 1948, California (Berkeley). Numerical analysis, curve fitting and numerical solution of differential equations.


Hellmut Goeke, Professor of Computer Science and Adjunct Professor of Electrical Engineering (1959). Diplom-Ing., 1953, Technische Hochschule Munich; M.S., 1955, Ph.D., 1959, Stanford. Programming languages, programming systems, compilers.


Richard F. Ladner, Associate Professor of Computer Science (1971). B.S., 1965, St. Mary’s College of California; Ph.D., 1971, California (Berkeley). Theory of computation, computational complexity, design and analysis of algorithms, data structures, recursive function theory.


Jerre D. Noe, Professor of Computer Science and Adjunct Professor of Electrical Engineering (1968). B.S., 1943, California (Berkeley); Ph.D., 1946, Stanford. Operating systems, modeling, analysis, measurements, and simulation of computer systems.


Adjunct, Research, and Lecturer Appointments

Donald L. Adolphson, Associate Professor of Quantitative Methods. B.A., 1966, California (Berkeley); M.S., 1968, Ph.D., 1973, Wisconsin. Mathematical programming, stochastic models in operations research, computers in business.

Alan Burns, Research Assistant Professor. B.S., 1971, Reed; M.S., 1974, Ph.D., 1979, Stanford. Programming languages, artificial intelligence, computers in education.


Alistair D. C. Holden, Professor of Electrical Engineering. B.S., 1955, Glasgow; M.E., 1958, Yale; Ph.D., 1964, Washington. Artificial intelligence and applications to speech understanding, vision and computer aided design.

David L. Johnson, Professor of Electrical Engineering. B.S.E.E., 1948, Idaho; Ph.D., 1955, Purdue. Switching theory and logical design, models of learning, concept formation, man-machine interaction.

Victor Klee, Professor of Mathematics. B.A., 1945, Pomona; Ph.D., 1949, Virginia; B.Sc. (Hon.), 1965, Pomona. Linear programming, network optimization, combinatorics, convexity, functional analysis.

Pierre A. MacKay, Professor of Classics, Near Eastern Languages and Literature, and Comparative Literature. B.A., 1954, Yale; M.A., 1959, Ph.D., 1964, California (Berkeley). Multi-lingual text-editing and typesetting, especially Arabic script; graphics; peripheral design.

James S. Meditch, Professor and Chairman of Electrical Engineering. B.S.E.E., 1956, Purdue; M.S., 1957, Massachusetts Institute of Technology; Ph.D., 1961, Purdue. Computer communications networks, telecommunications, and optimization theory.


Visiting and Post-Doctoral Appointments


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 (94 credits)
   a. For a student in the College of Arts and Sciences:
      Proficiency requirement: Math 124, 125, 126
      Humanities from the College distribution list: 20 credits
      Social Sciences from the College distribution list: 20 credits (Num & SS lists from A&S Central Advising)
      Free Electives: 39 credits
   b. For a student in the College of Engineering:
      Math 124, 125, 126
      Humanities and Social Sciences: 30 credits with at least 10 in each (must be chosen from Coll. of Engr. List)
      Functional Techniques: 10 credits (see list in 114 Sieg)
      Free Electives: 39 credits

2. Preparatory Component (31 credits)
   Math 205, 238
   Physics 121, 122, 123
   Electronics: choose one from Physics 334, EE 306, EE 310, or EE 355
      (only CSci and EE majors may ask for 310 and 355 entry cards).
   10 credits of Natural Science, Business, or Engineering

3. Computer Science Core Component (42 credits)
   CSci 201, 241, 321, 322, 326, 341, 378, 470, Math 464, and 6 credits of CSci 498

4. Computer Science Elective Component (13 credits)
   A student may satisfy this requirement by taking courses on the Approved Computer Science Electives List (pp. 14-15 of this brochure) or by taking graduate courses in Computer Science.
   To graduate, a Computer Science major must receive a grade of at least 2.0 in each course in (3) and (4) above.

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 = General Education Component, P = Preparatory Component, and C = Computer Science Elective Component.

Entering Computer Science at Sophomore Level:

Freshman Year

Sophomore Year

Junior Year

Senior Year

Entering Computer Science at Junior Level:

Freshman and Sophomore years (90 credits)
   Math 124-5-6, 205, 238, Physics 121-2-3, CSci 201, CSci 241, and general education and preparatory component credits.

Junior Year
   Autumn: CSci 321 (3), CSci 378 (5), G/P/C (7), Winter: CSci 322 (3), CSci 341 (5), Electronics, see p. 6 (3), G/P/C (4), Spring: CSci 326 (5), G/P/C (10).

Senior Year

It is recommended that students augment their computer science major with concentrated work in some application area 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 Computer Scientists to apply their 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

   Enrollment is limited due to space, faculty, and budgetary limitations. These applicants who, in the judgment of the faculty, are most
likely to succeed in the program and contribute to the field will be admitted. Minorities and women are encouraged to apply.

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. (Most students admitted to the department for Autumn '80 had at least 3.5 GPAs among the courses listed in a. above.)

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. TRANSCRIPTS ARE ALSO NEEDED BY THE ABOVE DATES. Applicants must supply copies of transcripts from all universities and colleges in which they took courses they are using for application purposes. If applying to the University as a transfer or postbaccalaureate student, the applicant may request the transfer information letter from 114 SING (343-1695). UW STUDENTS MUST ARRANGE FOR THEIR LATEST UW TRANSCRIPTS TO BE SENT FROM THE TRANSCRIPTS OFFICE.

4. Selection Process

The Computer Science Undergraduate Admissions Committee will meet each quarter during the academic year 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. Applicants will be notified of their admission or denial by letter 2-3 weeks after the above deadlines. 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.

---

Computer Science Courses

(Schedules subject to change)

Undergraduate Courses

CSci 201 Introduction to Computer Science (5) ANPsSu
   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) ANPsSu
   Basic algorithms, programming techniques, and basic concepts of the structured high-level language Pascal. Prerequisite: 201 or permission of instructor or dept. advisor.

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 majors only.

CSci 322 Introduction to Formal Models in Computer Science (3) W
   Finite automata and regular expressions; context free grammars and pushdown automata; nondeterminism; Turing machines and the halting problem. Emphasis on understanding models and their applications and on rigorous use of basic techniques of analysis. Induction proofs, simulation, diagonalization and reduction arguments. Prerequisites: 241 and 321. CSci majors only.

CSci 326 Data Structures (5) Sp
   Sequential and linked allocation of linear structures; tables, arrays, 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 majors only.

CSci 341 Programming Languages (5) W
   Designed to make the student reasonably fluent in several radically different languages, such as LISP, SNOBOL, APL, ALCOL 60, Pascal, SIMULA 67, and others. Prerequisite: 241. CSci majors only.

CSci 373 Data Structures and Algorithms (3) Anp
   Fundamental algorithms and data structures for their implementation. Techniques for solving problems by programming. Sorting, searching, linked lists, binary search trees, balanced trees, hashing. Offered jointly with EE 373. Prerequisite: 241 or 445 or equivalent knowledge of Pascal. For non-CSci majors, no credit if 326 has been taken.

CSci 378 Machine Organization and Assembly Language (3) A
   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 majors only. (EE 371 accepted as a substitute.)
CSci 401 Introduction to Assemblers and Compilers (3) W
Fundamentals of assemblers, compilers, and interpreters; symbol tables; macro-processing; lexical analysis, syntax analysis, semantic analysis and code generation for general purpose programming languages. Joint with EE 401. Prerequisites: 326 plus 378, or EE 371 plus CSci/EE 373, or instructor's permission.

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. Prerequisites: 322 and 326.

CSci 431 Introduction to Theory of Computation (3) Sp
Models of computation, computable and non-computable functions, space and time complexity, tractable and intractable functions. Prerequisite: 322.

CSci 445 Computer Programming Laboratory (1) AW
For experienced computer programmers who want to learn Pascal quickly. Topics include the syntax and semantics of Pascal along with programming examples. Taught in a concentrated fashion during the first two weeks of the quarter. Several programming assignments will be given to turn in throughout the quarter. Prerequisite: significant programming experience and fluency in a high-level language such as ALGOL, FORTRAN, BASIC, COBOL, or PL/I. No credit for CSci 241 if 241 has been taken. Cr/no cr only.

CSci 451 Introduction to Operating Systems (3) W
Principles of multiprogramming systems. Process management, resource management, and file systems. Prerequisites: 325 or CSci/EE 373 or instructor's permission.

CSci 470 Computer Design (4) AW
Fundamental gate circuits are developed into large logic gates. The use of these structures in the design of central processing units, memories, and peripheral equipment is illustrated. For computer science majors. Prerequisite: 378 or instructor’s permission. Entry card required for registration, first priority to CSci seniors - all others must get instructor’s permission before requesting the entry card.

CSci 473 Introduction to Artificial Intelligence (3) Sp
Principal ideas and developments in Artificial Intelligence: LISP and SNOBOL as the basis of precise descriptions of AI processes; theorem proving, problem-solving methods; representation of knowledge in procedures, and in frames; natural language analysis and synthesis, inference and generation from conceptual representations. Prerequisites: 326 and 341, or CSci/EE 373.

CSci 498-, 499H- Senior Project (1-6, max. 6; Honors 1-9, max. 9) AW
An independent development, survey, or small research project completed by the student in an area of specialization. Objectives are: (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. The work normally extends over more than one quarter. Prerequisite: senior standing. For Computer Science majors only. Entry cards are required.

CSci 499 Reading and Research (1-24) AW
Available in special situations for advanced computer science majors to do reading and research in the field. Subject to approval of the undergraduate advisor and a Computer Science faculty member. Usable as a free elective, but not in place of a core course or Computer Science elective. Entry cards are required. Prerequisite: senior standing. Cr/no cr only.

Graduate Courses

CSci 500 Computers and Society (2) W
Study of the impact of computer technology on present and future societies; 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. Cr/no cr only. Prerequisite: graduate standing in computer science or instructor's permission. (Offered alternate years.)

CSci 501 Compiler Construction (3) Sp

CSci 502 Advanced Topics in Compiler Construction (3) A
Translator-writing systems, incremental compilation. Design of production compilers. Prerequisite: 501. (Offered alternate years.) Cr/no cr only.

CSci 505 Concepts of Programming Languages (3) Sp
Basic concepts in programming languages, data structures (arrays, records) types, patterns, environments, central evaluation, application, matching; relation to high-level machines. Prerequisites: CSci/EE 401 and a working knowledge of Pascal and LISP.

CSci 506 Formal Semantics (3) W
Basic formalisms in semantics including flowchart schema, recursive schema, fixedpoint semantics and the associated induction rules, lambda calculus and other interpretive models. Formal semantics 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.

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 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
Digital representation of analog signals. Frequency domain and z-transforms of digital signals and systems. Design of digital systems; IIR and FIR filter design techniques, Fast Fourier Transform algorithms. Sources of error in digital systems. Analysis of noise in digital systems. Offered jointly with EE 518. Prerequisite: knowledge of Fourier analysis techniques, graduate standing or instructor's 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 a credit/no credit basis only.

CSci 521 Design and Analysis of Algorithms I (3) W

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: 521.

CSci 531 Formal Languages and Automata (3) A
Formal models in Computer Science including context free grammars, finite automata, regular expressions, Turing machines, and pushdown automata. The fundamental concepts of nondeterminism, undecidability, and syntax and semantics are covered.

CSci 532 Complexity Theory (3) W
Space and time complexity on various models of computation such as Turing machines, random access machines and list processing machines. Hierarchies based on complexity, time and space bounded reducibility, NP-completeness and other complexity classes, and provably difficult problems.

CSci 533 Computability and Logic (3) Sp
Formal systems that characterize the notion of computation and model the notion of logical reasoning. Computability of recursively enumerable sets, the recursion theorem, and diagonalization methods. Logic includes first order predicate logic, nonstandard models, proof systems, the completeness theorem, and undecidable theories.

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 542 Central Processor Architecture (3) Sp
Several central processing units are examined at the gate level. Included are the logic structures of: 1/0 bus, memory bus, ALU, address modification, control logic, combinatorial and multiplex instructions, access priority, cycle stealing, etc. Prerequisite: 470.

CSci 543 Analytic Models of Computer Systems (3) W
Emphasizes the use of queueing network models as tools for analyzing computer systems. Topics include useful results from basic queuing theory, selection of performance measures, modeling methodology, data acquisition, computational algorithms for queueing network models, approximation techniques, and decomposability and hierarchical modeling. A realistic case study will be undertaken.

CSci 548 Computer Systems Architecture (3) W
Notations for describing computer systems. Powerful CPUs. Memory organization. Channels and I/O processors. Microprogramming. Stack computers. Array and pipe line processors. 470 to be taken concurrently or instructor's 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 instructor's permission.

CSci 557 Computer Graphics (3) A
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. (Offered alternate years.)

CSci 573 Artificial Intelligence I (3) A
Introduction to the use of the computer in problem solving. Survey of theorem proving, symbol manipulating, pattern recognition, and inductive problem-solving techniques. Computer models of human thought. Prerequisite: 508 or 510 or instructor's permission.

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 instructor's permission.

CSci 590 Special Topics in Computer Science (*) A WSp
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: instructor's permission.

CSci 600 Independent Study or Research (*) A WSp
Cr/no cr only.

CSci 700 Master's Thesis (*) A WSp
Cr/no cr only.

CSci 800 Doctoral Dissertation (*) A WSp
Cr/no cr only.

Approved Computer Science Electives
(Schedules subject to change) Majors must choose a minimum of 13 credits from this list, see page 6.

CSci 401 Introduction to Assemblers and Compilers (3) W
CSci 421 Introduction to Analysis of Algorithms (3) A
CSci 431 Introduction to Theory of Computation (3) Sp
CSci 451 Introduction to Operating Systems (3) W
CSci 475 Introduction to Artificial Intelligence (3) Sp

A student may also take graduate courses in Computer Science to satisfy the elective component. See the instructor to determine the background necessary for graduate-level courses.

The following courses offered by other departments are acceptable as Computer Science Electives. Check for prerequisites, course entry requirements, and schedules with the department offering the course.

Acct 440 Accounting Systems (3) WSp
EE 417, 418 Introductory Communication Theory I, II (A, 3) W, Sp
EE 475 Digital Electronics and Microprocessors (4) A WSp
EE 476 Logical Design of Digital Devices (3) WSp
EE 479 Microcomputer System Design (5) WSp
Geo 463 Intro to Computer Cartography (3) W
GeoG 463 Computer Cartographies (3) Sp
Ling 461, 462 Syntax (3, 3) A WSp

The University of Washington, as a standing policy, does not discriminate against individuals because of their race, color, religion, age, sex, national origin, handicap, or status as disabled veteran or Vietnam Era veteran. Any discriminatory action can be a cause for disciplinary action. This policy applies to all University programs and facilities including, but not limited to, admissions, educational programs, and employment. Such discrimination is prohibited by Titles VI and VII of the Civil Rights Act of 1964, Title IX of the Education Amendments of 1972, Sections 503 and 504 of the Rehabilitation Act of 1973, the Age Discrimination in Employment Act Amendments of 1978, Vietnam Era Veteran's Readjustment Assistance Act of 1974, and other federal and state statutes and regulations. Inquiries regarding the application of these laws and regulations to the University may be directed to the University's Equal Employment Office; to the Director, Seattle Regional Office, Office for Civil Rights, U. S. Department of Education; or to the Director, Seattle Area Office, Office of Federal Contract Compliance Programs, Department of Labor.