UNDERGRADUATE HANDBOOK FOR COMPUTER SCIENCE

Department of Computer Science
University of Washington
127 Sieg Hall, FR-35
Seattle, Washington 98195
(206) 543-1695
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. 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 Vocational Rehabilitation Act of 1973, Age Discrimination Acts of 1974 and 1975, 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 Officer or to the Director, Seattle Regional Office, Office for Civil Rights, U.S. Department of Health, Education and Welfare.

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 1975, 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.
The year in parentheses is the year the faculty member joined the University.

Jean-Loup Baer, Associate Professor of Computer Science (1969). Diplome d'Ingenieur, Grenoble; Doctorat 3e Cycle, 1963, Grenoble; Ph.D., 1968, California (Los Angeles). Parallel processing, systems architecture, scheduling theory.

David B. Decker, Associate Professor of Mathematics and Computer Science (1948). A.B., 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.


Adjunct Prof. of


Richard E. 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.


Adjunct, Research, and Lecturer Appointments

Associate

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

George E. Dierh, Associate Professor of Quantitative Methods. Data base management systems, mathematical programming.

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

Alistair D. C. Holst, 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 Else, 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.

Associate

John S. Sobolewski, Research Professor of Computer Science and Medicine; Director, John L. Locke, Jr., Computer. Computer organization, computer communication, medical applications, information storage and retrieval.
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 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 (45 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:
       Math 124, 125, 126: 30 credits with at least 10 in each
       Functional Techniques: 10 credits
       Free Electives: 40 credits

(2) Preparatory Component (31 credits)
    Math 203, 238
    Physics 121-2-3, Engr 251
    Mathematics, Business, or Engineering (Math 391, 392, 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 (13 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 sophomore year. The number in parentheses indicates the number of credits; C, F, and G denote courses from the general education, preparatory, and Computer Science elective components respectively.

Entering Computer Science at Sophomore Level:

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

Sophomore Year
   Winter: Math 238 (3), G/P (9). Spring: G/P (15).

Junior Year
   Autumn: CSci 321 (3), Math 464 (3), G/P/C (9). Winter: CSci 378 (3), G/P/C (9).
   Spring: CSci 326 (5), G/P/C (9).

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

Entering Computer Science at Junior Level:

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

Junior Year
   Spring: CSci 251 (4).

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 area like Physics, Economics, Accounting, or Electrical Engineering. The 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, 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. TRANSSCRIPTS ARE ALSO NEEDED BY THE ABOVE DATES. UW STUDENTS MUST ARRANGE FOR THEIR LATEST TRANSCRIPTS TO BE SENT FROM ADMISSIONS AND RECORDS. Applicants must supply copies of transcripts from all universities and colleges they have attended in which they took courses they are using for application purposes.

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.

---

Computer Science Courses

CSci 201 Introduction to Computer Science (3) AWS
- A rigorous introduction to the theoretical and practical components of computer science: algorithms, programs, data structures, machines, computability, applications, social aspects. Prerequisite: MATH 126.

CSci 241 Programming (3) AWSpSU
- 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, 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 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, syntax analysis, semantic analysis and code generation for general purpose programming languages. Offered 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 (3) A
- Designed to make the student reasonably fluent in several radically different languages, such as LISP, SNOWBALL, APL, ALGOL 60, PASCAL, SIMULA 67, and others. Prerequisite: 378.

* Schedules subject to change.
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 design and structures in the design of central processing units, memories, and peripheral equipment is illustrated. For computer science majors. Prerequisite: 378 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. Theorems 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: 326 and 441.

CSci 478 Advanced Topics in Compiler Construction (3) W

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

CSci 505 Concepts of Programming Languages (3) Sp
Basic concepts in programming languages, data structures (arrays, records, types), patterns, environments, control, evaluation, application, matching, relations 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, fixpoint 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 programs 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 recursive 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 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 I (3) W
CSci 522 Design and Analysis of Algorithms II (3) 0
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, pushdown store automata. General parsing methods: top-down and bottom-up. Restricted parsing methods: LL(k), LR(k), precedence grammars, shift reduce parsing.

CSci 531 Syntaxic Analysis (3) A

CSci 532 Models of Computation (3) W

CSci 537, 538 Computability Theory (3,3) A
Introduction to the theory of effective computability. Formulation of the concepts of recursive and partial recursive function and functionally 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 sets, unsolvable problems, the arithmetical hierarchy, the relation of the degree of unsolvability of a set to the arithmetical hierarchy, the relation of the degree of unsolvability of a set to the arithmetical hierarchy, the relation of the degree of unsolvability of a set to the arithmetical hierarchy, the relation of the degree of unsolvability of a set to the arithmetical hierarchy, the relation of the degree of unsolvability of a set to the arithmetical hierarchy, the relation of the degree of unsolvability of a set to the arithmetical hierarchy.

CSci 540 Discrete System Simulation (3) A
Principles of simulation of discrete, event-oriented systems. Model construction, simulation and validation, and relationship to other systems. Use of special-purpose techniques for system simulation and analysis. Use of special-purpose techniques for system simulation and analysis. Use of special-purpose techniques for system simulation and analysis. Use of special-purpose techniques for system simulation and analysis. Use of special-purpose techniques for system simulation and analysis. Use of special-purpose techniques for system simulation and analysis.

CSci 541 Computer Measurement and Evaluation Techniques (3) Sp

CSci 542 Central Processor Architecture (3) Sp
Central processor architecture. Several central processing units are examined at the gate level. Several central processing units are examined at the gate level. Several central processing units are examined at the gate level. Several central processing units are examined at the gate level. Several central processing units are examined at the gate level. Several central processing units are examined at the gate level.

CSci 543 Analytic Models of Computer Systems (3) W
Emphasizes the use of queuing network models as tools for analyzing computer systems. Topics include useful results from queuing theory, selection of performance measures, modelling methodology, data acquisition, computational algorithms for queuing network models, approximation techniques, and decomposability and hierarchical modelling. A realistic case study will be undertaken. Prerequisites: 478 and 541, or permission.

CSci 548 Computer Systems Architecture (3) W

CSci 551 Operating Systems (3) Sp
Operating systems design and construction techniques. Systems programming languages, concurrent programming, design methodologies, deadlock problems, virtual memory allocation, and other topics. Study of the structure of different kinds of operating systems. Prerequisite: 541 or 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. Prerequisite: 500.

CSci 573 Artificial Intelligence I (3) A
Introduction to the use of the computer in nonsensory problem solving. Survey of theorems proving, symbolic manipulation, and neural network models. Prerequisite: 478.

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 (*) AWP
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 (*) AWSpS Credit/no credit only.
CSci 700 Master's Thesis (*) AWSpS Credit/no credit only.
CSci 800 Doctoral Dissertation (*) AWSpS Credit/no credit 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. Check for prerequisites and course entry requirements with the department offering the course.

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 405 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 (5)

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