DEPARTMENT OF COMPUTER SCIENCE
228 Roberts Hall, Phone: 543-1695
UNIVERSITY OF WASHINGTON

Undergraduate Degree Program in Computer Science

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 an M.S. and Ph.D. program since 1967. Starting in Autumn 1975, the department will offer a program of studies leading to the degree of Bachelor of Science. The curriculum will be introduced so that the first Bachelor of Science can be awarded in June, 1978.

Attached to this announcement is the following additional information on the Computer Science undergraduate program:

1. Catalog description
2. Course lists
3. Admissions policy
4. Tentative course forecast.

Applications are now being accepted for the program. Please call or write the Computer Science Department if you have any questions.

For general information the graduate course list is included.

CATALOG DESCRIPTION

COMPUTER SCIENCE

Computer Science is devoted to the representation, storage, manipulation, and presentation of information, in both theory and practice. Computer scientists are interested in: (a) representations of information, (b) algorithms to transform information, (c) languages to express algorithms, (d) hardware and software processors to execute algorithms, and (e) ways to accomplish all of these economically.

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; especially i portant is a foundation that will not become obsolete as technology advances and changes.

The computer field has a broad base of industrial and governmental computer users, providing many of the jobs suitable for the B.S. graduate. Typical jobs are systems analyst, systems programmer, technical salesperson, and 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.

Faculty

Jerre D. Noe, Chairman; Baer, Fischer, Herriot, Kehl, Ladner, Ritchie,
Alan C. Shaw, undergraduate program advisor through Autumn Quarter '75,
Hellmut Golde, undergraduate program advisor beginning Winter Quarter '76,
David B. Dekker, graduate program advisor
Adjunct, Research and Lecturer appointments; Adolphson, Diehr, Gillespie, Goldstein, Holden, D. L. Johnson, Klee, Rockafellar, Sobolewski

Undergraduate Program

Bachelor of Science

The Computer Science program is associated with both the College of Arts and Sciences and the College of Engineering. A student typically enters the program during the sophomore year or at the beginning of the junior year.

The required Computer Science curriculum consists of four components:

(1) General Education Component (80 credits)

A student may satisfy this component by completing 80 credits of approved general education courses. For example, a student in the College of Arts and Sciences may satisfy this component by study in the following areas:

Humanities from the College distribution list: 20 credits.
Social Sciences from the College distribution list: 20 credits.
Free electives: 40 credits.

A student in the College of Engineering may satisfy this component by study in the following areas:

Humanities and Social Sciences: 30 credits with at least 10 in each.
Functional Techniques: 10 credits excluding courses from the area of Computational Techniques.
Free Electives: 40 credits.

(2) Preparatory Component (46 credits)
Math 124-5-6, 205, 238 (also recommended Math 391, 392)
Phys 121-2-3
EE 351, 310
and 6 credits of Natural Sciences, Business or Engineering.

(3) Computer Science Core Component (40 credits)
CSci 201, 241, 321, 326, 378, 431, 441, 445-6-7, 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 Computer Science courses or other approved courses. Examples are courses in operating systems, compiler design, computer architecture, theory of computation, numerical mathematics, applications of computers to other fields, operations research, and management techniques.

Admission to the program. Selection of applicants will be made primarily on the basis of scholastic achievement and potential. Women and minorities are encouraged to apply. For more information, the Computer Science undergraduate advisor or the Arts and Sciences or Engineering advising centers should be consulted. It is recommended that the prospective applicant take Math 124-5-6, 205, 238, Phys 121-2-3 and CSci 201 during the freshman and sophomore years.

A sample curriculum program for a student who starts a Computer Science major during the sophomore year is shown below. G, P and C denote courses from the general education, preparatory and Computer Science elective components respectively.

Freshman Year
Autumn: Math 124 (5 credits); G/P(10) Winter: Math 125 (5); Phys 121 (4); CSci 201 (3); G/P (3). 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); CSci 378 (3); G/P (9). Spring: G/P (15).

Junior Year
Autumn: EE 351 (4); CSci 321 (3); G/P (8). Winter: Math 464 (3); EE 310 (3); G/P (9). Spring: CSci 326 (3); CSci 431 (3); CSci 470 (4); G/P (5).

Senior Year
Autumn: CSci 441 (3); CSci 445 (2); G/P (10). Winter: CSci 498 (3); CSci 446 (2); G/P/C (10) Spring: CSci 498 (3); CSci 447 (2); G/P/C (10).

DEPARTMENT OF COMPUTER SCIENCE

Courses for Undergraduates

CSci
201 Introduction to Computer Science (3) W
The purpose of this course is to provide a rigorous introduction to the theoretical and practical components of Computer Science: algorithms, programs, data structures, machines, computability, applications, social aspects. This course may be considered both as a vehicle for motivation and a survey. Prerequisite, Mathematics 124.

CSci
241 Programming (3) A
This course is designed to teach some basic algorithms and programming techniques, as well as the 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, Mathematics 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 (3) 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
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, 321 and 241.

CSci
441 Programming Languages (3) A
This course is designed to make the student reasonably fluent in several radically different languages. The course will highlight significant concepts from languages such as LISP, SNOBOL, APL, ALGOL 60, PASCAL, SIMULA 67, and others. Corequisite, 445.

CSci
445, 446, 447 Programming Languages Laboratory (2,2,2) AWSp
Practical programming experience in a number of computer languages, such as FORTRAN, ALGOL, COBOL, SNOBOL, LISP, APL, and assembly language. The laboratory will be based on independent study, supplemented by occasional lectures. Students must register for three consecutive quarters. 445 corequisite, 441.

CSci
498 Senior Project (1-6) WSp
The senior project consists of a report (and perhaps demonstration) describing a development, survey, or small research project completed by the student in an area of specialization. The project has the objectives of: (a) applying and integrating the classroom material from several courses, (b) introducing the professional literature, (c) gaining experience in writing a technical document, and (d) 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.

UNIVERSITY OF WASHINGTON

DEPARTMENT OF COMPUTER SCIENCE

Winter 1975

Admission Policy for the Undergraduate Program in Computer Science

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 judgement 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:

Mathematics 124, 125
Physics 121
an introductory computer science course, such as 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. Transfer students must first contact the Office of Admissions and obtain the necessary forms for applying to the University College of Arts and Sciences or College of Engineering.

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 pre-selection requirements (Section 2 above) may apply. Applications may be obtained from the Computer Science Department.

Completed applications must be received by the department by the 4th week of the quarter before the quarter of desired entrance (Spring for Autumn; Autumn for Winter; Winter for Spring).

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 the appeals procedure can be obtained from the chairman of the admissions committee.

March, 1975

DEPARTMENT OF COMPUTER SCIENCE
UNDERGRADUATE PROGRAM
TENTATIVE COURSE FORECAST

Title	Autumn	75	76	77	Winter	76	77	78	Spring	76	77	78
201 Intro. to Computer Science					MWF 9:30	X	X	X				
241 Programming	MWF 10:30	X	X	X								
321 Discrete Structures	MWF 9:30		X	X								
326 Data Structures									MWF 11:30		X	X
378 Mach. Org. & Assem. Lang.					MWF 3:30		X	X				
431 Theory of Computation I									MWF 2:30			X
441 Prog. Languages	MWF 1:30			X								
445,6,7 Language Lab.	TTh 1:30			X	TTh 1:30			X	TTh 1:30			X
498 Senior Project					*			X	*			X

C Sci 470 Computer Design (4) A,W
Fundamental gating circuits are developed into logic gating structures. The use of these structures in the design of central processing units, memories, and peripheral equipment is illustrated. Course for majors in Computer Science. Prerequisite, permission of instructor.

C Sci 472 Computer Software Systems (3) W
Principles of operating systems, compilers, assemblers, interpreters, and loaders for digital computers. Not intended for graduate students in Computer Science or Electrical Engineering with emphasis on advanced programming. Not open to students who have taken 501. Offered jointly with the Department of Electrical Engineering as Electrical Engineering 472. Prerequisite, 478.

C Sci 478 Computer Organization and Machine Language Programming (4) A,Sp
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. Offered jointly with the Department of Electrical Engineering as Electrical Engineering 478. Prerequisites, Engineering 141 and 346, or equivalent.

C Sci 501 Compiler Construction I (3) A
Basic concepts and design of interpreters and compilers. Lexical analysis, syntax analysis, storage management, and code generation for general-purpose languages. Offered jointly with the department of Electrical Engineering as Electrical Engineering 501. Prerequisite, 478 and working knowledge of a block structured programming language.

C Sci 502 Compiler Construction II (3) W
Advanced topics in compiler construction. Translator writing systems, incremental compilation, compiler-interpreters. Practical considerations for production compilers. Offered jointly with the Department of Electrical Engineering as Electrical Engineering 502. Prerequisite, 501.

C Sci 504 Comparative Study of Programming Languages (3) Sp
There are several major traditions in the development of programming languages. Important programming languages from various traditions will be studied and compared so that major contributions of each language will be understood. Possible languages for study include: TRAC, LISP 1.5, ALGOL 60, PASCAL, SIMULA 67, SNOBOL 4, and APL. Prerequisite, 478.

C Sci 505 Concepts of Programming Languages (3) W
Programming languages are a means of communicating between man and machine. The notions of high level language and high level machine are examined. Major concepts in language, such as data structures (arrays, records, patterns, etc.), control structures (procedures, coroutines, processes, etc.), and environment structures are studied. Examples from existing languages are given to illustrate concepts. Prerequisite, 478.

C Sci 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 flowcharts; path length on trees. Garbage collection. Dynamic storage allocation. Data management on external media. Prerequisite, 478 or permission of the instructor.

C Sci 510 LIST PROCESSING AND STRING MANIPULATION (3) W

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.

C Sci 520 COMPUTER SCIENCE SEMINAR (1) A,W,Sp

Weekly discussion by students and faculty or visitors on current topics of interest.

C Sci 531 AUTOMATA THEORY I (3) W

Phrase-structure grammars. Relation between grammars and automata. Deterministic and nondeterministic finite state automata, pushdown store automata, linear-bounded automata, and Turing machines.

C Sci 532 AUTOMATA THEORY II (3) Sp

Continuation of 531. Selected topics from closure properties of languages, time and tape bounded Turing machines, deterministic pushdown automata, LR(k) grammars, decision problems, abstract complexity theory, and other topics in automata theory. Prerequisite, 531.

C Sci 537, 538 COMPUTABILITY THEORY (3,3) A,W

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 arithmetical 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, Mathematics 502 or permission.

C Sci 540 DISCRETE SYSTEM SIMULATION (3) W

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 their functional components and data structures. Prerequisite, programming experience with ALGOL.

C Sci 541 COMPUTER MEASUREMENTS AND EVALUATION TECHNIQUES (3) Sp
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.

C Sci 542 CENTRAL PROCESSOR ARCHITECTURE (3) W
Several central processing units will be 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.

C Sci 548 COMPUTER SYSTEMS ARCHITECTURE (3) W
Notations for describing computer systems. Powerful CPU's. Memory organization. Channels and I/O processors. Micro-programming. Stack computers. Array and pipe-line processors. Prerequisite, 478. Co-requisite, 470 or permission.

C Sci 551 OPERATING SYSTEMS (3) Sp
Design and construction principles for multiprogramming systems: Early batch processing systems. Cooperating sequential processes. Multiprogramming. Main storage management. Process and resource control and scheduling. File systems. Projects using the Computer Science Laboratory. Prerequisite, 508.

C Sci 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 will be required to complete a project on the interactive graphics facility in the Computer Science Laboratory. Prerequisite, 508.

C Sci 573 ARTIFICIAL INTELLIGENCE I (3) W
Introduction to the use of the computer in non-numerical problem solving. Survey of theorem proving, symbol manipulating, pattern recognition, and inductive problem-solving techniques. Computer models of human thought. Prerequisite, 478.

C Sci 574 ARTIFICIAL INTELLIGENCE II (3) Sp
Continuation of studies of artificial intelligence systems emphasizing theorem proving, symbolic problem solving, pattern recognition, and natural language data processing. Students will be required to do projects. Prerequisite, 573 or permission of instructor.

C Sci 590 SPECIAL TOPICS IN COMPUTER SCIENCE (*) A,W,Sp
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. May be repeated for credit. Prerequisite, permission.

C Sci 600 INDEPENDENT STUDY OR RESEARCH (*) A,W,Sp,S

C Sci 700 MASTER'S THESIS (*) A,W,Sp,S

C Sci 800 DOCTORAL DISSERTATION (*) A,W,Sp,S