FOR SENIORS AND GRADUATE STUDENTS

200. Systems Programming. (3) Review of batch process systems programs. Implementation techniques for parallel processing of input-output and interrupt handling. Addressing techniques, core management, file system design and management, system accounting, and other user-related services. Overall design of operating systems. Multiprogramming systems on multi-processor hardware. Traffic control, interprocess communication, design of system modules, and interfaces. System updating, documentation and operation. Prerequisites: Courses 150, 160, and 170, or equivalent.

210. Introduction to Automata Theory. (3) Structure, decompositions, mappings, and applications of sequential machines and other automata. Prerequisites: Course 110 or 211, and Mathematics 158; or equivalent.

211. Applied Boolean Algebra. (3) Boolean algebra with applications to switching circuits, sets, and logic. Prerequisite: Math 11 or 22, or equivalent.

212. Mathematical Logic. (3) Propositional calculus. First order theories and model theory. Elementary arithmetic and Godel's incompleteness theorems. Prerequisite: Math 165 or 191, or equivalent evidence of mathematical maturity. (Same as Mathematics 212).

213. Recursive Function Theory. (3) Effective computability and Church's thesis, unsolvability results, recursively enumerable sets, reducibilities, finite and infinite extension arguments, applications, priority arguments, and hierarchy theory. Prerequisite: Course 212. (Same as Mathematics 213).

216. Formal Language Theory. (3) Representation, classification, properties and grammars of formal languages. The relation of formal languages to automata. Decidability questions. Prerequisite: Course 210 (may be taken concurrently, with permission of the instructor).

220. The Computer as Instrumentation for Research in the Humanities and Social Sciences. (3) Topics and problems chosen in relation to the research interests of the seminar members, with special attention to new applications possibilities relevant for social scientists and humanists. Prerequisite: Course 120 or permission of instructor.

223. Computational Linguistics. (3) Preparation of dictionaries and concordances; distribution studies; parsing algorithms for context-free, context-sensitive, and transformational grammars. Prerequisites: Course 16 or 120 and Linguistics 101. (Same as Linguistics 283.)

225. Computational Stylistics. (3) Computational approaches to the study of style in language and literature. Stylistic discrimination by culture, genre, and author, with attention to problems in stylistic development. Survey and use of computer program packages. Prerequisite: Course 16 or 120, or equivalent. (Same as Linguistics 285).

230. Simulation of Cognitive Processes. (3) An introduction to "creative" information processing systems including examples of individual and coordinated human behavior, and artificially intelligent computer programs. Elementary systems analysis, simulation techniques and heuristic programming as needed to study cognitive processes. Construction and validation of a simulation of some "intelligent" information-processing system. Prerequisites: Courses 150 and 160; Course 235 also desirable.

235. Systems Simulation. (3) Introduction to Monte Carlo methods and their applications. Random variable generation, queuing, and variance reducing techniques. Definition and use of special languages for discrete simulation. Selected examples of simulations in business, engineering, industry and science. Prerequisites: Course 100 and Mathematics 127, or consent of instructor.

255. Information Retrieval. (3) Information analysis, dictionary construction, and automatic information systems. Dictionary, statistical and syntactic operations. Retrieval models and processes. Input-output systems including auxiliary services. Evaluation of retrieval systems. Prerequisite: Course 150.

260. Programming Structures. (3) Relation of programming structures to data and machine structures, static and dynamic structures, recursive and iterative structures, program and command structures, microprogram and macroprogram structures, and extension of language and data structures. Prerequisite: Course 200.

265. Compiler Construction. (3) Review of program language structures, translation, loading, execution, and storage allocation. Compilation of simple expressions and statements. Organization of a compiler including compile-time and run-time symbol tables, lexical scan, syntax scan, object code generation, error diagnostics, object code optimization techniques, and overall design. Use of compiler writing languages and bootstrapping. Prerequisites: Courses 150 and 160, or equivalent.

270. Computer System Design. (3) Study of such problems as arithmetic and non-arithmetic processing, error detection and handling, memory utilization and hierarchy, storage management, addressing, control, interrupt handling, and input-output including graphics. Comparison of alternate solutions as implemented in actual systems. Selected new approaches to computer system organization. Prerequisites: Course 170; Course 200 and EE 287 are also desirable.

280. Numerical Analysis of Linear Systems. (3) Computational aspects of linear algebra, linear equations and matrices, direct and indirect methods, eigenvalues and eigenvectors of matrices, error analysis. Prerequisites: Course 180 and Math 190.

281. Numerical Functional Analysis. (3) Foundations of functional analysis: linear spaces, convergence and completeness, operators in Hilbert space. Non-linear operators: iterative methods, fixed point theory, error estimation, monotonicity theory. Applications: boundary value problems, eigenvalue problems, approximation theory, solution of non-linear equations, solution of differential and integral equations. Prerequisites: Math 175 and Math 182 or concurrent enrollment(s).

290. Advanced Topics. (1-3) Arranged as needed to present appropriate material to senior and graduate students. May be repeated for additional credit. Prerequisite: Variable.

294. Human Factors in Computer-Based Systems. (3) Human characteristics relevant for machine design and software design, and for task and job definition; performance control in program and system generation and implementation; social organization characteristics affecting system acceptance and functioning. Prerequisite: Course 170 (Course 200 desirable).

295. Seminar. (1-3) Discussion of and reports on current literature in computer science. May be repeated for additional credit. Prerequisite: Variable.

299. Special Problems. (1-3) Arranged to allow senior and graduate students to pursue work in special problem areas under individual guidance by the staff. May be repeated for additional credit. Prerequisite: Variable.

The Ph.D. Program
in Computer Science
at the
University of Kansas
and
Kansas State University

FOR GRADUATE STUDENTS

310. Theory of Automata. (3) Formal models for computation and their applications, with emphasis on algebraic and logical aspects. Prerequisite: Course 210 or consent of instructor.

314. Theory of Computability. (3) Turing machines, equivalent notions of effective computability, unsolvability results, and computational complexity. Prerequisite: Course 210 or 212, or consent of instructor.

316. Advanced Formal Language Theory. (3) Language manipulation algorithms, recognition, generation and translation strategies. Generalized automata and structure manipulation. Syntax, semantics, interpretation, redundancy, and ambiguity. Complexity of language computation. Other topics as time permits. Prerequisite: Course 216 (Course 314 desirable).

381. Numerical Solution of Nonlinear Operator Equations. (3) Iterative methods for the solution of nonlinear problems: Newton's method and Newton-like methods, generalized secant method, Davidon methods, continuation methods. Convergence analysis: Kantorovich's theorem, error estimation, rates of convergence. Function minimization and nonlinear least squares. Prerequisite: Course 281.

382. Numerical Solution of Ordinary Differential Equations. (3) One step and multistep methods for initial value problems. Stability, consistency and convergence of these methods. Techniques for boundary value problems. Stiff ordinary differential equations and special stability problems. Prerequisite: Math 181.

383. Numerical Solution of Partial Differential Equations. (3) The numerical solution of hyperbolic, parabolic, and elliptic equations by finite difference methods; discretization and round-off errors; the concept of stability for initial value problems; the solution of elliptic boundary value problems by variational and projection methods. Prerequisites: Courses 280 and 281.

385. Numerical Functional Approximation. (3) General finite interpolation, convergence theorems for interpolatory processes, uniform approximation, best approximation, approximation in normed spaces, approximation of linear functionals, spline functions. Prerequisite: Course 281.

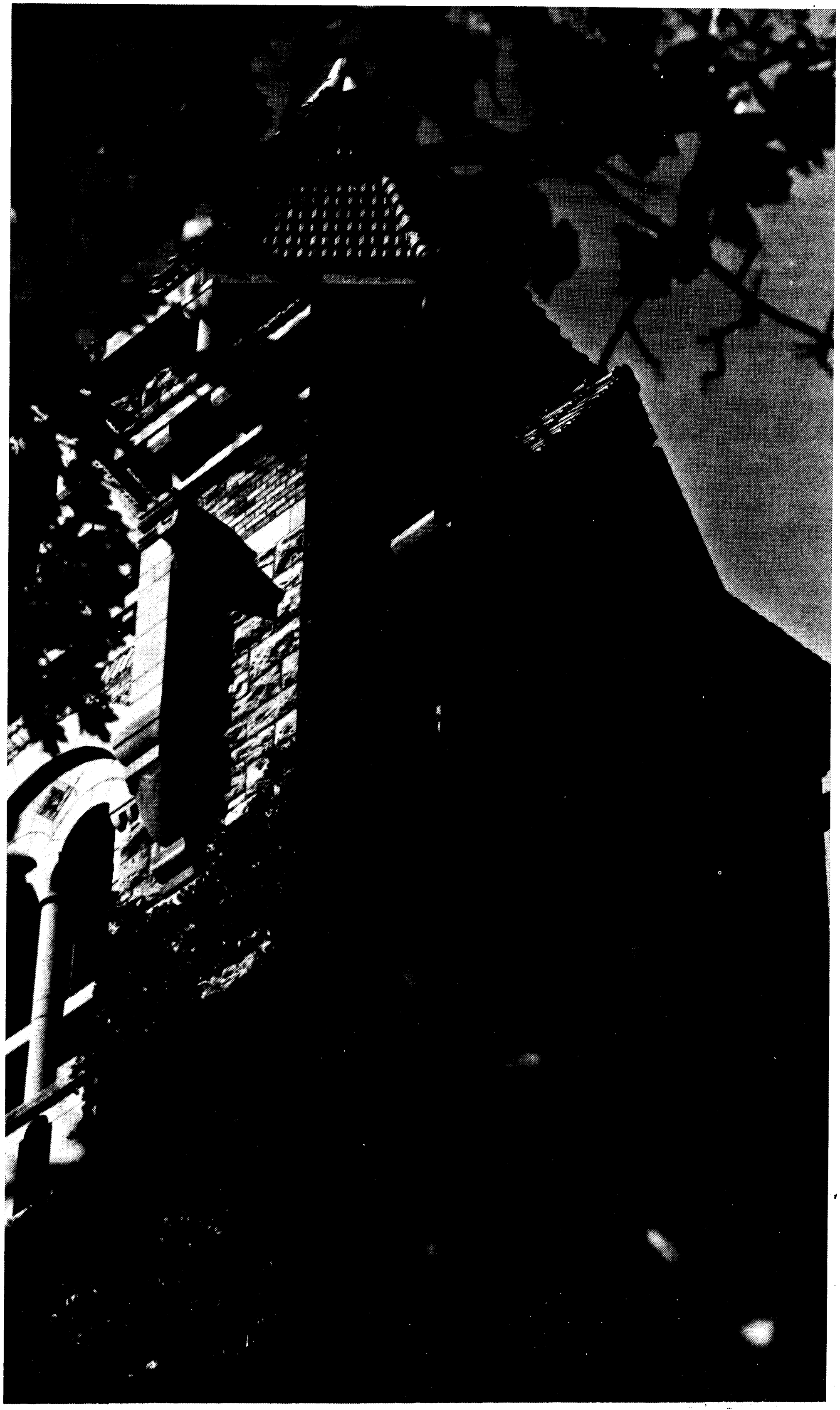
387. Computational Statistics. (3) Computational solution of statistical problems emphasizing distribution functions, special functions, approximations, error analysis, pseudo-random numbers, and verification of random processes. Prerequisites: Courses 180 and 235, and Math 332 or 333.

390. Graduate Topics. (1-3) Arranged as needed to present appropriate material to graduate students. May be repeated for additional credit. Prerequisite: Variable.

395. Research Seminar. (1-3) One or more current research areas in computer science are investigated. Each student will prepare one or more research papers as well as presenting his work orally. May be repeated for additional credit. Prerequisite: Variable.

398. Master's Thesis. (1-6).

399. Doctoral Thesis. (1-10).



Courses of Instruction

AT KANSAS STATE UNIVERSITY

Faculty: Ahmed, Brewer, Calhoun, Conrow, Fisher, Gallagher, Miller, Sackman, Sincovec, Trump, Unger, Weinberg.

FOR UNDERGRADUATE CREDIT ONLY

315. Fundamentals of Computer Programming. (3) Introduction to a procedure-oriented language, the description of a digital computing system, the strategy of problem solving using a digital computer, and the concepts and properties of algorithms. Applications to problem solving. Prerequisite: High School Algebra.

397. Seminar in Computer Science. (arranged)

FOR UNDERGRADUATE AND GRADUATE CREDIT

Note: Courses numbered between 400 and 599 may not be counted toward a graduate degree in Computer Science.

400. Introduction to Algorithmic Processes. (3) Introduction to algorithms, language and notation for describing algorithms, analysis of computational problems and development of algorithms for their solution. The notation of lists, tables, data sets (files) and records. Prerequisite: Course 315.

410. COBOL. (3) Elements of data processing in the COBOL language. Applications. Prerequisite: Course 315.

425. Computer Organization and Programming I. (3) Logical organizations of computers; number systems and arithmetic control units and instruction sequencing, assemblers, addressing systems, subroutine linkages (transfer vectors), and input-output operations. Prerequisite: Course 400, or consent of instructor.

440. Introduction to Programming Languages. (3) Structure of algorithmic languages. Conversational languages. List processing and string manipulation languages. Concepts and facilities of programming languages. Prerequisite: Course 400.

505. Mathematical Machines and Computability I. (4) Elements of matrix algebra pertinent to digital computations. Computer methods of solving linear equations and inverting matrices. Error analysis, problem conditioning and post-optimizations. Rectangular and singular systems. A generalized inverse for matrices. Algorithmic methods of solving eigenvalue problems, progressive algorithms. Applications. Three hours lecture, one hour laboratory on a digital computer. Prerequisite: Math. 221 and Course 315.

506. Mathematical Machines and Computability II. (4) Computer algorithms for finding roots of polynomials and the real roots of transcendental equations. Error analysis, effect of uncertainty in the coefficients. Computer algorithms for the approximation of continuous functions. Numerical integration, differential equations. Three hours lecture and one hour laboratory on a digital computer. Prerequisite: Course 505 or Math. 223 and Course 315.

525. Computer Organization and Programming II. (3) Study of information representation and processing techniques. Transformations between storage media. Referencing of information as related to the structure of its representation and implications for the design of the referencing language. Prerequisite: Course 425.

535. Non-numeric Programming. (3) The use of computers in areas not involving numeric calculations. Surveys of applications into areas such as music, learning theory, games and discrete pattern recognition. Heuristic programs. Prerequisite: Course 400.

597. Seminar in Computer Science. (arranged)

600. Discrete Structures. (3) Study of linear lists, strings, arrays, orthogonal lists, and graphs. Representation of such structures within computers. Prerequisite: Course 400.

610. List and String Processing Languages. (3) The languages LISP and SNOBOL will be studied, and problems will be developed illustrating the use of each. Prerequisite: Courses 400 and 600.

615. Theory of Computability. (3) Propositional calculus, axiomatics. Turing machines; unsolvable problems; quantification theory; satisfiability and validity models, first-order theories; foundational considerations. Prerequisite: E.E. 355, or consent of instructor.

620. Programming Systems. (3) Languages for writing software, design of assembly systems, macro-instructions, operating systems (monitors), interrupt systems, storage allocation, and multiprogramming. Prerequisite: Courses 400 and 600.

631. Numerical Solution of Ordinary Differential Equations. (2) (Concurrent with Math 631) Computer algorithms and techniques for solving ordinary differential equations. Programming exercises on the digital computer. Prerequisite: Math 555, Computer Science 315 or Computer Science 505, Math 240 plus concurrent enrollment in Math 631.

632. Numerical Solution of Partial Differential Equations. (2) (Concurrent with Math 632) Computer algorithms and techniques for solving partial differential equations. Programming exercises on the digital computer. Prerequisite: Math 631, Computer Science 631 plus concurrent enrollment in Math 632.

635. Artificial Intelligence. (3) Application of heuristics to problem solving. Perceptrons, pattern recognition, learning, self-evolving programs. Prerequisite: Course 535.

640. Programming Languages. (3) The study of the structure and facilities of major algorithmic procedure-oriented languages and their implementation, operation and use. Prerequisite: Courses 440 and 600.

670. Information Organization and Retrieval. (3) Models for representing structured information, techniques for organizing and searching files. Structure of semiformal languages. Analysis of information by statistical, syntactic and logical methods. Applications to automatic information retrieval systems, question answering systems, and man-machine interaction. Prerequisite: Course 425.

701. Automata Theory. (3) Finite automata; synchronous sequential circuits; Kleene's Theorem; semi-groups; monomorphisms; generator systems; algebraic linguistics; potentially infinite machines; theory of computability, recursive functions; programming systems. Prerequisite: Course 615 and Math. 512, or consent of instructor.

710. Compiler Design I. (3) Formulation of syntax-directed and table-driven techniques used in compiler design. Various alternative techniques. Environment of a compiler, conversational compilers. Prerequisite: Courses 620 and 640.

711. Compiler Design II. (3) Conversational compilers. Syntax-directed compilers. Extensible compilers. Compiler writing systems. Prerequisite: Course 710.

712. Seminar in Computer Science. (1)

720. Business Data Processing. (4) Manual, semi-automatic, automatic systems of data processing. Accounting concepts, data processing implications. Organization of sequencing and direct-access files. Checking and control techniques. Student groups will study business applications and recommend data-processing systems. Three hours lecture, two hours lab each week. Prerequisite: Courses 410 and 600.

760. Computers, Science and Society. (3) Critical review of computers and social problems. Impact of computers on science and experimental method. Guided research on some aspect of the social use of computers selected by each student. Prerequisite: Consent of the instructor.

797. Seminar in Computer Science. (arranged)

FOR GRADUATE STUDENTS

800. Computational Semantics. (3) Theoretical prerequisites and computational techniques for mechanical interpretation of language sentences. Semantics of formal computer languages, including query languages for information retrieval. Structural representation of meaning. Prerequisite: Course 615.

810. Computer Simulation. (3) A variety of examples will be studied to illustrate the power and flexibility of automata, theoretic, representation, statistical techniques, and information theory in simulation studies. Prerequisite: Course 620.

811. Computer Simulation Experiments. (3) A computer simulation will be programmed for the digital computer and used to predict data and test hypotheses. Prerequisite: Course 810, or consent of instructor.

815. Special Topics in Computer Science. (2-4) Study in selected areas of artificial intelligence, computational linguistics, linear and nonlinear programming, theorem proving by computer, models of intelligent processes, and the like. Prerequisite: Consent of instructor.

865. Computer Simulation of Eco-systems. (3) A selection of various eco-systems will be used to show how a computer can be used to formally define an eco-system. Then computer simulated eco-systems will be used to discover ways to optimize the benefits to be derived from actual eco-systems. Prerequisite: Course 810, or consent of instructor.

998. Research in Computer Science and Mass Communication. (arranged)