Computer Science at the University of Kentucky
An Undergraduate Handbook

February 2006

Department of Computer Science
773 Anderson Hall
University of Kentucky
Lexington, KY 40506-0046

Tel (859) 257-4997
e-mail: martha@cs.uky.edu
www: http:\\www.cs.uky.edu

I. INTRODUCTION

The Computer Science Department offers a Bachelor of Science (BS), Master of Science (MS), and Doctor of Philosophy (Ph.D.). The Department was founded in the 1960's and is located in the College of Engineering.

There are 28 faculty members and 50 graduate assistants associated with the department. We teach approximately 35 different courses each year and conduct an active colloquium series. There are about 375 undergraduate majors. Undergraduate majors are subject to a selective admissions policy as detailed in the University Bulletin.

The program leading to an undergraduate degree in Computer Science is a demanding one that attracts intelligent, hard working students. Currently, the employment opportunities for graduates of the program are excellent with all our graduates, who elect industrial positions, finding positions offering promise for personal and professional growth. Many of our graduates also pursue advanced degrees sponsored either by industry or supported by graduate fellowships and assistantships.

Students may participate in our CoOp program that provides alternate semesters of academic study and full-time employment. Details are available from:

Engineering Co-op Program
367 Ralph G Anderson Bldg
University of Kentucky
Lexington, KY 40506-0503

There are close ties between the Departments of Computer Science Department, Computer and Electrical Engineering, and Mathematics as demonstrated by several joint faculty appointments and cross-listed courses.

This handout introduces the faculty and undergraduate courses offered by the department. Information about the graduate program is included in the Graduate Handbook available from the department.

II. ADMISSION

The University of Kentucky has differing admissions requirements and procedures for in-state residents, out-of-state U.S. residents, foreign students, and transferring students. Community college students should plan to transfer to the Lexington campus at the start of their second year. Students waiting until their junior year to start completing their major requirements will find that they will need more than four semesters at the Lexington Campus. Application materials and information about financial aid can be obtained from the University web site: http://www.uky.edu or by writing the Computer Science Department.

The Computer Science program has a lower-division component and an engineering-standing component. Admission to the University of Kentucky is sufficient for admission to the lower-division component of the program.

Applicants seeking admission to engineering standing will be judged on their performance in the lower-division courses in both computer science and mathematics. Details appear in the University Bulletin.

III. FACULTY AND RESEARCH INTERESTS

Anthony Q. Baxter, Associate Professor, Ph.D. 1973, University of Virginia. Programming Systems, Performance Monitoring and Evaluation, Database Systems.

Kenneth L. Calvert, Associate Professor, PhD 1991, University of Texas at Austin. Computer Network Protocols, Network Security

Fuhua Cheng, Professor, Ph.D. 1982, Ohio State University. Computer Graphics, Numerical Analysis, Computer-Aided Geometric Design.

Alexander Dekhtyar, Assistant Professor, PhD 2000, University of Maryland. Uncertainty in Databases and AI, Multi-agent Systems, Data Mining.

Craig Douglas, Professor, PhD 1982,, Yale University. Computational Sciences, Parallel Computing, Numerical Analysis.

Zongming Fei, Assistant Professor, PhD 2000, Georgia Institute of Technology. Computer Networking, Telecommunications, Distributed Multimedia Systems.

Raphael A. Finkel, Professor, Ph.D. 1976, Stanford University. Operating Systems, Distributed Algorithms, Programming Languages.

Judy Goldsmith, Professor, Ph.D. 1988, University of Wisconsin-Madison. Structural Complexity, Computational Geometry.

James Griffioen, Associate Professor, Ph.D. 1991, Purdue University. Computer Networks, Operating Systems.

Jane E. Hayes, Assistant Professor, PhD 1999, George Mason University. Software Verification and Validation, Software Engineering, Information Technology.

Robert Heath, Associate Professor, Ph.D. 1973, Auburn University. Computer Engineering, Digital Signal Processing, Software Engineering. (Joint with Electrical Engineering)

Jerzy W. Jaromczyk, Associate Professor, Ph.D. 1984, Warsaw University. Analysis of Algorithms, Complexity Theory, Graphics.

Christopher Jaynes, Assistant Professor, PhD 1998, University of Massachusetts. Computer Vision, Artificial Intelligence.

Amit Kale, Assistant Research Professor, PhD 2003, University of Maryland. Computer vision, pattern recognition, and video processing.

Debby Keen, Lecturer, PhD 1994, University of Kentucky. Computer Science Education.

Andrew Klapper, Professor, Ph.D. 1982, Brown University. Cryptography, Spread Spectrum Communications.

Kenneth K. Kubota, Professor, Ph.D. 1969, Facultes des Sciences de Paris. Number Theory, Operating Systems. (Joint with Mathematics)

Forbes D. Lewis, Professor, Ph.D. 1970, Cornell University. Theory of Computation, Computational Complexity. (Emeritus)

D. Mannivannan, Assistant Professor, Ph.D. 1997, The Ohio State University. Distributed Computing Systems, Database Systems, Operating Systems

Victor W. Marek, Professor, Ph.D. 1968, Warsaw University. Logic, Logic Programming, Artificial Intelligence, Database Systems.

Ryan McKenzie, Lecturer, MS 2005, University of Kentucky**.** Computer Science Education.

A. C. R. Newbery, Professor, Ph.D. 1962, University of London. Numerical Analysis, Interpolation. (Emeritus)

David Nister, Assistant Professor, PhD 2001, Royal Institute of Technology (KTH). Computer vision, video processing and graphics in general, including reconstruction, recognition, comprehension, search and interaction based on visual input, advanced rendering and visualization.

Paul Piwowarski, Lecturer, MS 1982, University of Kentucky, MS in EE, Michigan State University 1972. Computer Science Education**.**

Brent Seales, Associate Professor, Ph.D. 1991, University of Wisconsin. Image Processing & Graphics.

Mukesh Singhal, Professor, PhD 1986, University of Maryland. Wireless Network and Mobile Computing, High-Speed Computer Networks, Operating Systems

Miroslaw Truszczynski, Professor & Chair, Ph.D. 1980, Technical University of Warsaw. Logic Programming, Artificial Intelligence.

Gregorz W. Wasilkowski, Professor, Ph.D. 1980, Warsaw University. Complexity of Approximately Solved Problems, Numerical Analysis.

Ruigang Yang, Assistant Professor, PhD. 2003, University of North Carolina. Computer Graphics, Computer Vision, and Multimedia.

Jun Zhang, Associate Professor, Ph.D. 1997, The George Washington University. Scientific and Parallel Computing, Applied Numerical Analysis.

IV. COMPUTING FACILITIES

The CS Department owns and operates the MultiLab in Engineering Annex 202, the UnderLab in CRMS 215E, and the HandsOn lab in EE 102. The MultiLab has 30 workstations (2.7GHz Intel P4, 256MB, 30GB). The UnderLab has 33 workstations (1.86GHz Intel P4, 512MB, 80GB). The HandsOn lab has 16 workstations (1.5GHz Intel P4, 128MB, 20GB). The MultiLab and UnderLab computers are dual-boot; students may reboot these machines to run either Linux or Windows, or (in the MultiLab) a student-created Linux kernel. Each instructional lab has associated servers that provide file service to Linux via NFS and to Windows via Samba.

Students can access the machines in the MultiLab via the network when the machines are running Linux. During academic semesters, the labs are opened at least five days a week for console access.

Students in our classes also use the University-run CSLab in Engineering Annex 203, which has about 20 dual-2GHz Macintosh Power G5 computers running Mac OS X. The Lab is dedicated to the needs of CS students.

The University runs several Windows XP laboratories around campus that are available to our students.

V. COURSES

The following are commonly offered undergraduate courses. Unless noted otherwise courses are 3 credit hours, meeting for 3 50-minute or 2 75-minute periods per week.

CS-100 The Computer Science Profession (1)
An introductory seminar which covers the fundamental activities, principles, and ethics of the computer science profession. An overview of the discipline of computer science, examples of careers, the history of computing, and experience with elementary computing tools are included.

CS 101 Introduction to Computing I
An overview of computing for non-majors.

CS 115 Intro. to Computer Programming
This course teaches introductory skills in computer programming using an object-oriented computer programming language. There is an emphasis on both the principles and practice of computer programming. Covers principles of problem solving by computer and requires completion of a number of programming assignments

CS 215 Intro. to Program Design (4)
This course introduces students to object-oriented design and problem solving. Course subjects include data structures, dynamic data and pointers, and recursion. There is an introduction to sorting, searching, and the complexity of algorithms. Prereqs: CS-115

CS 221 A 1st Course in CS for Engineering Majors (2)
An introduction to computing for non-Electrical Engineers majors.

CS 216 Introduction to Software Engineering
Software engineering topics to include: life cycles, metrics, requirements specifications, design methodologies, validation and verification, testing, reliability and project planning. Implementation of large programming projects using object-oriented design techniques and software tools in a modern development environment will be stressed. Prereq: CS 215.

CS 275 Discrete Mathematics (4)
Topics in discrete math aimed at applications in Computer Science. Fundamental principles: set theory, induction, relations, functions, Boolean algebra. Techniques of counting: permutations, combinations, recurrences, algorithms to generate them. Introduction to graphs and trees. Prereqs: MA-113 and CS-115

EE 280 Design of Logic Circuits
Boolean algebra, combinational logic circuits, synchronous sequential circuits; asynchronous sequential circuits; design problems using TTL integrated circuits. Prereq: CS-222 (or 115).

CS 315 Algorithm Design and Analysis
Introduction to the design and analysis of algorithms. Asymptotic analysis of time complexity. Proofs of correctness. Algorithms and advanced data structures for searching and sorting lists, graph algorithms, numeric algorithms, and string algorithms. Polynomial time computation and NP-completeness. Prereq: CS-215, CS-275, and engineering standing.

CS 316 Web Programming
This course introduces students to the World Wide Web, languages and techniques used for web programming, data transfer over the Internet, and the tools available in the web environment. Prereqs: CS-216

CS 321 Introduction to Numerical Methods
Floating point arithmetic. Numerical linear algebra; elimination with partial pivoting and scaling. Polynomial and piecewise interpolation. Least squares approximation. Numerical integration. Roots of equations. Ordinary differential equations. Laboratory exercises using software packages. Prereq: (i) MA 114 and (ii) CS 221. Knowledge of a procedural computer language is required. (Same as MA 321).

CS 335 Graphics and Multimedia
This course focuses on the graphical human-machine interface, covering the principles of windowing systems, graphical interface design and implementation, and processing graphical data. There is an emphasis on medium-scale programming projects with graphical user interfaces using a high-level procedural programming language and concepts such as object-oriented design. Prereq: CS-216 and engineering standing

CS 375 Logic and Theory of Computing

Topics in logic and discrete math aimed at applications in Computer Science. Propositional calculus: truth tables, logical relations, proofs, tautologies, soundness. Predicate calculus: variables, quantifiers, equivalences. Models of computation: logic circuits, finite automata, and Turing machines. Prereq: CS 215, CS 275, and engineering standing.

CS/EE 380 Microcomputer Organization

Hardware and software organization of a typical computer; machine language and assembler language programming, interfacing peripheral devices, and input-output programming; real-time computer applications, laboratory included. Prereq: EE 280. (Same as EE 380.)

CS 395 Independent Work in Computer Science (2)

A course for computer science majors only. A problem approved by the Chairman provides an opportunity for individual research and study. May be repeated to six credits. Prereq: major and a gpa of 3.00 and consent of instructor.

CEP 399 Cooperative Education (1)

A course designed for undergraduate students in the Co-op program. The course may be taken on a P/F basis only. Prereq: Prior approval from the Director of Co-op Education.

CS 405G Introduction to Database Systems

Study of fundamental concepts behind the design, implementation and application of database systems. Brief review of entity-relationship, hierarchical and network database models and an in-depth coverage of the relational model including relational algebra and calculi, relational database theory, concepts in schema design and commercial database languages. Prereq: CS-315 and graduate or engineering standing.

CS 415G Graph Theory

Variations of graphs, properties of graphs, graph theory algorithms, and applications of graph theory to other disciplines. Prereq: consent of instructor. (Same as MA 415G)

CS 416G Principles of Operations Research I

This course is an introduction to modern operations research and includes discussion of modeling; linear, dynamic, and integer programming, scheduling and inventory problems, and network algorithms. Prereq: MA 213. (Same as MA 416G).

CS 422 Numerical Solutions of Equations

Linear equations: Gaussian elimination, special linear systems, orthogonalization, eigenproblem, and iterative methods. Nonlinear equations: solutions of equations in one variable, solutions of systems of nonlinear equations. Optimization. Prereq: CS/MA 321 and MA 322; or consent of instructor. (Same as MA 422).

CS 441G Compilers for Algorithmic Languages

The techniques of processing, specifying, and translating high-level computer languages are studied. Topics include finite state machines and lexical analysis, context-free grammars for language specification, attributed translation grammars, language parsing, and automatic generation of compilers by SLR, LALR, and other methods for analyzing context-free grammars. Other topics may include code optimization, semantics of programming languages, and top-down parsing. Prereq: CS-315 and engineering standing.

CS 450G Fundamentals of Programming Languages

An intensive study of fundamental programming concepts exhibited in current high-level languages. Concepts include recursion, iteration, coroutines, multiprocessing, backtracking, pattern-matching, parameter passing methods, data structures, and storage management. Object oriented languages and their supporting run-time environment are covered. Prereq: CS-315 and engineering standing.

CS 463G Artificial Intelligence

The course covers basic techniques of artificial intelligence. The topics covered in this course are: search and game-playing, logic systems and automated reasoning, knowledge representation, intelligent agents, planning, reasoning under uncertainty, and declarative programming languages. The course covers both theory and practice, including programming assignments that utilize concepts covered in lectures. Prereqs: CS-315, CS-375, and engineering standing.

CS 470G Introduction to Operating Systems

This course provides an introduction and overview of operating system design, internals, and administration. Topics include classical operating systems (process management, scheduling, memory management, device drivers, file systems), modern operating systems concepts (kernel/microkernel designs, concurrency, synchronization, interprocess communication, security and protection), and operating system administration. Prereq: CS 315, CS-380, and engineering standing.

CS 471G Networking and Distributed Operating Systems

Broad overview of concepts in networking and distributed operating systems with examples. Topics will include protocol stacks, link, network, transport, and application layers, network management, the client-server model, remote procedure calls, and case studies of distributed OS and file systems. Prereq: CS-315 and engineering standing.

CS 480G Advanced Computer Architecture

This course focuses on advanced computer architectures and low-level system software. Topics include RISC architectures, vector and multiprocessor architectures, multiprocessor memory architectures, and multiprocessor interconnection networks. Peripheral devices such as disk arrays, NICs, video/audio devices are covered. Topics also include device drivers, interrupt processing, advanced assembly language programming techniques, assemblers, linkers, and loaders. Prereq: CS/EE-380 and graduate or engineering standing.

CS 485 Topics in Computer Science

Studies of emerging fields in Computer Science. A review and extension of selected topics in the current literature. May be repeated to a maximum of six credits under different subtitles. Prereq: Engineering Standing.

CS 499 Senior Design Project

Projects to design and implement complex systems of current interest to computer scientists. Students will work in small groups. Prereq: CS-315 and engineering standing

CS 505 Advanced Concepts in Database Systems

The course introduces a variety of modern techniques in database and distributed database systems. The major topics are object-oriented database systems, distributed databases, heterogeneous databases, and knowledge based systems. Also included is a discussion of logic in databases and the logical equivalence of various logical and database languages). The prime objective of this course is to teach a variety of methods that would allow one to apply database techniques to the solution of database problems in areas where traditional relational database methods are not viable. Prereq: CS-405 and engineering standing.

CS 515 Algorithm Design

The design and analysis of efficient algorithms and data structures for problems in sorting, searching, graph theory, combinatorial optimization, computational geometry, and algebraic computation. Algorithm design techniques: divide-and-conquer, dynamic programming, greedy method, and randomization, approximation algorithms. Prereq: CS 315 and engineering standing.

CS 521 Computational Sciences

Study of computer science techniques and tools that support computational sciences and engineering. Emphasis on visualization, performance evaluation, parallel computing, and distributed computing. Prereq: CS 115, CS/EE 380, and engineering standing.

CS 537 Numerical Analysis

Floating point arithmetic. Direct methods for the solution of systems of linear algebraic equations. Polynomial and piecewise polynomial approximation, orthogonal polynomials. Numerical integration: Newton Cotes formulas and Gaussian quadrature. Basic methods for initial value problems for ordinary differential equations. The emphasis throughout is on the understanding and use of software packages for the solution of commonly occurring problems in science and engineering. Prereq: CS/MA 321 or equivalent, or graduate standing, or consent of instructor. (Same as MA 537 and EGR 537). Knowledge of procedural computer language is required.

CS 541 Advanced Compiler Design

A study of the theory and practice of implementing compilers for high-level language with emphasis on the use of compiler-compilers for automatic generation of compiler systems. Topics include specification of languages by grammars, LR, SLR, LALR, and LL parsing algorithms, lexical analysis, syntax directed translation, code optimization and generation, and data flow analysis. Prereq: CS 441 or CS 575.

CS 535 Intermediate Computer Graphics

Three-dimensional graphics primitives such as 3D viewing, lighting, shading, hidden line/surface removal, and more advanced topics such as solid modeling, image storage and representation, advanced raster graphics architecture and algorithms, advanced modeling techniques, and animation will be covered. Prereq: CS-335, CS-315, CS-321, and engineering standing.

CS 555 Logic for Computer Science

The course covers fundamentals of propositional and predicate logic, and their uses in declarative programming to model and solve computational problems. Topics include propositional satisfiability, satisfiability testing techniques such as the DPLL algorithm, automated reasoning techniques for predicate logic such as resolution with unification and logic programming. Prereqs: CS 315 and CS 375 or consent of instructor.

CS 563 Artificial Intelligence

This course is primarily concerned with general problem-solving methods: production systems, graph searching, and automated theorem-proving, in particular, the resolution and its variants. Topics include heuristics, games on trees, and minimax methods, as well as a study of various knowledge-representation schemes such as frames, prototypes, predicate logic and basic methodology of expert systems. Prereq: CS 315, CS 375, and engineering standing.

CS 570 Modern Operating System

Brief review of classical operating system concepts (process and memory management, process coordination, device drivers, file systems, starvation/deadlock). Modern topics of file systems (log-structured file systems, distributed file systems, memory-based file systems), operating system design (monolithic, communication-kernel, extensible/adaptable, distributed shared memory), multiprocessor issues (scheduling, synchronization, IPC), security (internet attacks, encryption, defenses). Inspection and modification of actual operating system code (Linux). Prereq: CS 470 and engineering standing.

CS 571 Computer Networks

Principles of computer networks using current Internet technologies and protocols as examples. Routing algorithms and protocols; end-to-end transport; flow control; congestion avoidance and control; mail, web, and file transfer protocols; designing and implementing applications using common network APIs. Advanced topics, included as time permits, include network security, multicast, and quality of service. Prereqs: CS 471G or consent of instructor.

CS 575 Theoretical Aspects of Computing

The formal study of computation, including computability and computation with limited resources. Church's thesis and models of computation. Formal languages and machines as recognizers of languages. The Chomsky Hierarchy of language types. Topics may include Turing machines or other basic models of computation; decidability and undecidability; basic complexity theory; finite automata and regular languages; pushdown automata and context-free languages. The course will cover primarily theory, including assignments that utilize concepts covered in lectures. Prereqs: CS 375 or consent of instructor.

CS 585 Intermediate Topics in Computer Science

Topics to be selected by staff. May be repeated to a maximum of six credits but only three credits may be earned by a student under the same topic.

CS 587 Microcomputer Systems Design

A course in the design of microcomputer systems for hardware engineers which includes the following topics: use of uncommitted logic arrays in instruction set design; hardware support for operating systems and programming languages; customizing microcomputers for specific execution environments; and control of concurrency. Prereq: EE 581 and EE 583 or consent. (Same as EE 587)

VI. DEGREE REQUIREMENTS

The Computer Science program includes courses dealing with the design, implementation, analysis, and software-engineering issues related to algorithms and computer programs. A foundation in continuous and discrete mathematics is used to study numerical problems and to analyze algorithms. Through required and elective courses students are exposed to the fundamentals of computing theory and algorithms, programming languages, language translation and compiling, graphics, scientific computing, artificial intelligence, networks, databases, and operating systems.

The degree requirements for a BS include: University Studies, College, and Departmental requirements. The University requirements are detailed in the University Bulletin. The College requirements are incorporated into the departmental requirements. A minimum of 128 credit hours is required to graduate. Transfer students must complete at least 30 credits at the University of Kentucky and at least 24 credits must be applicable for the Computer Science major.

The department recognizes the desire of students to combine computer science with other disciplines. Technical electives afford the opportunity to take courses in other areas. We find that students often concentrate their electives to obtain a minor in Mathematics or Business and Economics. Potential employers or graduate schools will favorably react to students having as many courses as possible in the sciences (i.e., Physics, Chemistry, etc.), in Mathematics and Statistics, and in Engineering as well as in their Computer Science major.

Students must attain engineering standing prior to taking most junior and senior-level courses. The requirements for attaining engineering-standing are the completion of CS-100, CS-115, CS-215, CS-216, EE-280, ENG-104), MA-113, MA-114, PHY-231, and PHY-241 with a minimum cumulative GPA of 2.5 in these courses. University repeat options may be utilized as appropriate.

Degree Requirements – BS in Computer Science

The following curriculum meets the requirements for a B. S. in computer science.

Freshman Year	
First Semester	**Hours**
CS-100 Computer Science Profession	1
CS-115 Computer Science I	3
ENG-104* Writing or Natural Sciences Elective [N]	3-4
MA-113* Calculus I	5
University Studies [U]	3
Second Semester	
CS-215 Computer Science II	4
ENG-104* Writing or Natural Sciences Elective [N]	3-4
MA-114 Calculus II	5
University Studies [U]	3

Junior Year	
First Semester	**Hours**
CS-380 Microcomputer Organization (same as EE-380	3
CS-321 Numerical Analysis (same as MA-321)	3
University Studies [U]	3
ENG-2xx Writing Intensive Course	3
Free Elective [E]	3
Second Semester	
CS-375 Logic and Theory of Computing	3
CS elective [C]	3
Technical elective [T]	3
University Studies [U]	3
Natural Science [N]	3
Writing Intensive Course	3

Sophomore Year	
First Semester	**Hours**
CS-216 Intro. to Software Engineering	3
CS-275 Discrete Mathematics	4
MA-213 Calculus III or MA-322 Linear Algebra	3-4
PHY-231* General University Physics	4
PHY-241 General University Physics Lab	1
University Studies [U]	3
Second Semester	
EE-280 Design of Logic Circuits	3
CS-315 Algorithm Design and Analysis	3
PHY-232* General University Physics	4
PHY-242 General University Physics Lab	1
STA-281 Probability & Statistics	3
University Studies [U]	3

Senior Year	
First Semester	**Hours**
CS-470 Intro. to Operating Systems	3
CS elective [C]	3
Technical elective [T]	3
University Studies [U]	3
Free Elective [E]	3
Second Semester	
CS-499 Intro. to Senior Design Project	3
CS elective [C]	3
Technical elective [T]	3
Technical elective [T]	3
Free Elective [E]	3

* - satisfies a University Studies requirement

[U] - To be selected from University Studies areas in Social Sciences, Humanities, Cross-Cultural, and Electives in conjunction with the academic advisor.

[N] - Any natural science course excluding more elementary versions of completed required courses.

[C] - Computer Science electives include 300 level and above computer science courses with two to be selected from: CS 335, CS 405G, CS 441G, CS 450G, and CS 463G.

[T] - Technical electives include any 300 level and above courses in computer science, electrical engineering, mathematics, and business and economics. MA 214 is also an acceptable technical elective.

[E] - Two courses must be in areas other than computer science, science, engineering, or mathematics to satisfy the University Studies Program and the computer science ABET accreditation requirements. Any remaining electives should be selected to meet the minimum total of 128 hours required for graduation.

Minor in Computer Science

Students obtaining a degree in another major can obtain a minor in computer science in conjunction with their degree by completing the following computer science courses:

Course	Hours
CS-115	3
CS-215	4
CS-216	3
CS-275	4
CS-315	3
CS elective	3