2006 Computer Science Department Self Study

Computer Science Department
Personnel and Budget Committee

Jennifer Whitehead, Chair
Yung Kong
Christopher Vickery
Jerry Waxman
Zhigang Xiang

April, 2006

Table of Contents

Introduction

This self-study is being undertaken during a period of active ferment that places our discipline in the center of the global shift to the information age and all that this shift implies for institutions of higher education. As liberal arts institutions such as Queens College adapt their curricula to the still-emerging nature of digital information technology in the lives and careers of our students, computer science plays a core, arguably unique, role in the evolution of these curricular shifts.

We begin by placing our department's context within the gamut of fields that currently fall under the "computer" rubric. Our presentation here is based on the structures presented in a report called *Computing Curricula 2005* [ACM 2005], which was produced jointly by three professional organizations: the Association for Computing Machinery (ACM), the Association for Information Systems (AIS), and the Computer Society of the Institute of Electrical and Electronics Engineers (IEEE-CS).

When our department was founded in 1971, and for another two decades or so after that, "computing" was taught in one of three distinct flavors: electrical engineering dealt with hardware, computer science dealt with software, and information systems dealt with automated business practices. But these fields have matured and expanded to the point that there are now five identifiable categories of undergraduate computer-related degree programs:

1. Computer Engineering
2. Computer Science
3. Information Systems
4. Information Technology
5. Software Engineering

Each of these five categories has its own particular focus, although they overlap to various extents in the particular topics they deal with. Broadly speaking, Computer Engineering emphasizes hardware design and implementation, Software Engineering emphasizes software design and implementation, Information Systems deals with the information processing and communication needs of businesses and other organizations, and Information Technology deals with the implementation and maintenance of organizations' networking and computing infrastructure. We'll discuss Computer Science in a moment.

> To make the above distinctions more concrete, consider Queens College as an organization conducting its "business" of providing education to our students. The Office of Converging Technologies (OCT) relies on the Information Technology skills of its employees. They provide and maintain the networking, communication, and computing backbone for the campus at large. Next, the College has a number of offices that rely heavily on the Information Systems skills of their employees: the Registrar's office and the Bursar's office to name just two. The College also deploys software in a wide variety of contexts, from professors' individual course web pages to online registration to library services. All this software, whether developed informally by individuals or purchased from commercial software producers, depends on the skill sets that fall under the rubric of Software Engineering for the appropriateness of their designs and the robustness of their implementations. Finally, the computers and other hardware devices used in the College were designed by computer engineers.

As we mentioned above, the topics dealt with by the various disciplines overlap significantly. Computer engineers need to understand software in order to develop the hardware platforms that will support software systems; information technologists need to understand networking hardware as well as the software that controls it; information systems designers need to understand networking, computing devices, and software system capabilities to be effective.

Programs of study in the various disciplines are typically offered by different types of higher education institutions. The distinctions are not hard and fast, but at least two of the cases are rather easy to identify: computer engineering is found in institutions with other engineering programs, and information systems programs are offered by institutions with business-oriented programs. For example, within the City

University, City College and Baruch College fall into these two categories respectively. The other two disciplines discussed so far have less clear cut “home bases.” Information technology can be thought of as an outgrowth of applied programs that have heretofore drawn on the skill sets provided by two year programs such as the Electrical Technology program at Queensboro Community College. As the field has matured, four year programs in this field are emerging to address the expanding skill set demands of the area.

Software engineering covers perhaps the broadest range of computing topics of any of the five discipline areas identified by the Overview Report. Because it deals with software, this discipline must make contact with hardware in one direction. Furthermore, because software engineering deals with issues regarding the management of large software projects, it also extends to the areas of “business practices” in the other direction.

So, where does this leave the computer science discipline, and why is computer science the most appropriate one of these five disciplines for a liberal arts college such as Queens College?

The four disciplines discussed so far all focus closely on practical aspects of computing: the design of computing equipment, the design of software systems, the design of computer-based business systems, and the deployment of computer and communication infrastructures. What distinguishes a computer science curriculum from these others is the attention it gives to the theoretical underpinnings of computational processes. That is, the emphasis is more on the principles of computing than on its practice. Principles and practice are not mutually exclusive, and one of the goals of our department is in fact to graduate students who are well-qualified to work in technical positions. But by emphasizing principles in our course work, we aim to do far more than just train students for their first job. Our goal is to provide students with a solid basis for dealing with and adapting to the particular forms computing-related technology takes over the graduate’s career span. We don’t know the exact form these technological advances will take over the years and decades ahead, but we do know that the principles of information encoding, algorithmic analysis, and computational structures (both hardware and software) that we teach will continue to provide the basis for emerging digital technologies for the foreseeable future.

The traditional goal of a liberal arts college has been to expose students to the key elements of the humanities, social sciences, and natural sciences in order to develop their skills as critical thinkers. By developing these skills, the liberal arts graduate can not only participate productively in the work force, but also handle leadership positions in their areas of interest as well. With the transition to an economic base that depends at least as much on services and information management as it does on production and delivery of hard goods and services, the traditional liberal arts education must expand to include technological literacy and competence as a core element. We submit that as the College continues the process of reviewing its core curriculum structure, it might well also consider revising its ten-year old mission statement to say “ ...to prepare students to become leading citizens of an increasingly global and technology-driven society ... .”

With their emphasis on the *principles of digital technology,* computer science departments nationwide are squarely at the center of the expanding definition of a liberal arts education. The role of the Computer Science Department is central to the core mission of Queens College.

Mission Statement

With the foregoing in mind, our department's mission can be stated as follows:

> The mission of the Computer Science Department is, primarily, to provide instruction and to conduct research in the core areas of computer science: software design, theoretical foundations, and hardware systems. Regarding instruction, our courses for computer science majors are designed (a) to provide knowledge and skills that will enable our graduates to immediately enter the workforce as productive employees in the technology sector of the economy, and (b) to provide an understanding of the basic principles of computer science that will serve as a strong foundation for expanding and evolving their knowledge and skills as their careers progress.
>
> Another aspect of our mission is to teach computing-related courses that serve as important components of the liberal arts curriculum of the College. In this regard, we are actively involved in cooperative programs with other departments, and provide service courses and minors for students who need instruction in various areas of applied computing, such as information technology and information systems.
>
> The Department also offers a master's degree program which serves computing professionals and others with bachelor's degrees who wish to extend their knowledge of computer science. Additionally, the Department is involved in preparing the next generation of computer science researchers by participating in the CUNY Ph.D. Program in Computer Science, which is housed at the Graduate Center.

Resources

Current State: Department

Human

In addition to the faculty, the department has a staff of five full-time employees:

- One Higher Education Associate. Network administrator and lead member of the technical support team.
- One Higher Education Assistant.
- One College Assistant
- Two CUNY Office Assistant L-2

Physical

Site Summary: The department is centered in the "A" wing of the Science Building. The department office is on the second floor, along with the chair's office, six faculty offices, an office for the HEA/HEa, an office for shared use by adjuncts, an office used by PhD graduate students, three student labs, a conference room, a networking closet, a storage room, and three mixed/shared-use faculty labs. Eight faculty offices are located on the first floor, and four more are located on the third floor.

The department is also the primary user of a computer-equipped classroom in SB B-131, and is currently completing the installation of an additional student networking lab in A-103.

Laboratories: The department maintains specialized instructional labs for Hardware Design, Computer Graphics, Operating Systems, and Networking. The equipment in the Hardware Design Lab was purchased using funds from an NSF grant in 2003, and consists of eighteen high-end PCs with specialized software for logic circuit design, and a similar number of FPGA (Field Programmable Gate Array) prototyping systems. The other labs have been equipped primarily using funds from the University's Technology Fee charged to all students each semester.

The department is in the process of converting a classroom into an instructional lab for web design using Tech Fee funds, and maintains mixed instructional/research labs in the areas of bioinformatics, genetic algorithms, information retrieval, mobile communications, and wireless technologies.

Computer and Networking Infrastructure: The network is the lifeblood of the department, and we are fortunate to have the technical staff capable of designing, installing, maintaining, and managing this vital resource.

The network connects switches, printers, and over 200 computers in the department office, faculty offices, and student and research laboratories. Computers and software cover a wide variety of systems, from PCs running Windows or OS-X to workstations running Solaris or Linux. The network has been designed to isolate student-accessible computers from faculty and staff machines, while providing all systems with access to the Internet through a firewalled link to the campus network. The network supports a mix of Gigabit, 100 Megabit, and some legacy 10 Megabit connections.

The department maintains a variety of network services for instructional, research, and professional purposes. These include networked file systems and printers, as well as servers for standard Internet protocols such as HTTP, SSH, and SMTP.

Fiscal

The department receives funding from the following sources:

- Tax-Levy Funds

  OTPS/TS (Other Than Personnel Services / Temporary Services)

  Our OTPS budget for 2006-06 was $20,688, which includes $10,000 for an office assistant, $2,400 in PSC travel funds, temporary service money for tutors, and the remainder for equipment, supplies, and Xeroxing. In addition, two new hires (Reddy and Zheng) received start-up funds of $85,000 and $70,000 respectively.

  Released Time Account

  The department has an account with a current balance of approximately $48,000 to cover certain summer salaries and teaching shortfalls due to released time taken by regular faculty. This account is funded from both internal grants such as PSC-CUNY awards and from external grants.

  Maintenance

  The Sun workstations in the department cost $12,614 to maintain annually. This cost is covered by funds provided by the dean of the division.

  Graduate Assistants

  The College provides the department with funds to hire one Graduate Assistant A to teach eight credits a semester and the CUNY Graduate Center provides one Graduate Assistant C to teach six credits a semester. The number of and funding source for Grad A and Grad C lines can vary from year to year. In Fall 2006 the department will also receive two Graduate Teaching Fellows from the CUNY Graduate Center who will be teaching 6 hours per semester. These graduate students have full responsibility for the courses they teach. There is no funding for traditional teaching assistants (TAs) in the department.

- Non-Tax-Levy Funds

  Grant Overhead

  The department has about $3,400 a year available from overhead recoveries associated with the department's grants and contracts.

  Tech Fee

  The University collects a Technology Fee in the amount of $75 per full-time student per semester. Although the College's share of the fees collected is large, none of it is explicitly allocated to the department. Rather, the department, like all other academic departments and administrative units, applies for funding of "projects" each year.

  Queens College Foundation

  The department is able to draw about $3,000 a year from this account, which is funded primarily by alumni donations.

  Computer Associates Adjunct Assistant Professor

  The department offers one course a semester that is taught by an Adjunct Assistant Professor who is employed by Computer Associates, Inc. His teaching salary is paid directly to him by CA. The nominal value of this service is approximately $6,000 per year.

Current State: College and CUNY

Library

Despite its close physical proximity to the Benjamin Rosenthal Library (BRL), the main library at the College, the department relies very little on the bricks and mortar facilities provided by that facility. Rather, the department relies primarily on access to on-line information resources for its research and instructional information access. The campus library provides the department with valuable access to online technical books through a subscription to the Safari Bookshelf. The College and University provide access to the online versions of many journals and monographs, including the Association for Computing Machinery's "Digital Library." However, neither the College nor the University provides access to the IEEE's Electronic Library (IEL), nor even the IEEE's Computer Science Electronic Library (CSLSP-e). The IEL is a resource that would be extremely valuable to our faculty.

Laboratories and Computers

The Computer Science Department is, of course, totally dependent on computing resources in virtually every aspect of its instructional and operational existence. While the department manages its own computers for faculty and specialized laboratories, it relies on the College for general computing resources for student use; our students make heavy demands on campus "computer labs" such as those in the Library, in the I Building, and on the first floor of the Science Building. (There are no college computer labs outside the department that are specifically designated for use by our students.)

With the price of a good desktop computer from brand-name manufacturers now less than $300 (probably less than the price of a semester's textbooks for a full-time student), we now assume students have access to their own computing equipment off campus. But not all students have laptops that they can or want to bring to the campus, so the college-supplied computers on campus remain important.

The Computer Science laboratories have benefited tremendously from Technology Fee money, which provides funds for upgrading department labs every three years. Software and licenses are also available for teaching purposes. In addition, funds have been allocated for a new lab and "smart classroom" focusing on web development and computer security. Renovation of classroom SB A103, which will house this lab, is scheduled to begin in Summer 2006.

Research laboratory space in the department is at a premium. There are three main laboratory spaces (Rooms 207 A, B, and C) intended for research, but in fact they are used both as laboratories and to house shared resources like network printers and servers.

Networking Infrastructure

The College provides wired and wireless networking for the campus through the Office of Converging Technologies (OCT). As mentioned previously, the department maintains its own internal network infrastructure, but relies on the campus network for its access to the Internet at large. The College recently upgraded the Department's link to the campus backbone as well as some of its internal switches to Gigabit Ethernet, making a great improvement in our local area network speed. In addition, the College has just completed wiring a dedicated T1 line for the department, allowing us to access this resource for teaching and research purposes where the College firewall might prevent research in certain risky areas, such as computer security and data integrity.

Wireless access on campus is quite good, but many of the classrooms used for our classes lack both wired and wireless access. This is a particularly painful situation for our courses in networking and web design, but should be remedied soon as Technology Fee funds will be used to upgrade all classrooms on campus to "smart classrooms" and the College adds wireless access points to the Science Building.

In 2004, the College ranked 13th nationally in Intel's first annual "Most Unwired College Campuses" survey. However the College's rank dropped to 45 in the current survey. As Intel notes:

> While last year many campuses had minimal wireless network deployment, this year's survey reveals that students are more likely to be enjoying the benefits of campus life, unwired. On average, 98 percent of the top 50 campuses are covered by a wireless network, up from 64 percent in 2004, with 74 percent having 100 percent wireless network coverage on campus, up from just 14 percent last year.

Intel, of course, is promulgating its products with these surveys, and there is no evidence we know of that correlates a college's wireless ranking with its academic standing. But our own networking infrastructure must integrate closely with the College's in order to provide Internet access to and from the department. OCT and the department must inter-operate smoothly for the department's network to operate effectively.

The College and University provide email accounts for faculty and students. These accounts are in the process of transitioning to a centralized LDAP system based on IBM's Lotus technology.

Effects of Current Resources

Our discussion at this point does not include issues related to faculty staffing levels and workloads. But the resource constraints mentioned above do affect both our teaching and research productivity negatively. Until classrooms are upgraded with appropriate technologies (networking, media projectors, and the other accouterments of smart classrooms), until faculty are supported in their teaching by traditional TAs, and until the library provides us with online access to missing resources like IEL, the department remains at a competitive disadvantage in terms of both productivity and our ability to attract high-caliber students.

While we stress the fundamental principles underlying technology in our courses, we constantly update existing courses and revise our curriculum to make sure the technologies we use as our examples are current. Current funding levels and the uncertainty of Technology Fee funding make it difficult for us to keep our course offerings in sync with our curricular changes.

Fortunately the cost of new technology tends to improve over time, although some of this may be an artifact of the recent era of low interest rate economics. As a result, we have been generally successful in funding and maintaining our instructional and research laboratory equipment. However, we are totally dependent on our own small technical staff for our very survival. We have learned that OCT is spread too thin to provide timely and effective support for our department's networking, computer, and software needs.

A significant gap in current library facilities, which negatively impacts our research productivity, is the lack of electronic access to key publications of the IEEE.

Current amounts of laboratory and office space are constraining the department. New hires are being forced to use offices that have no windows. Instructional labs serve double duty as part-time research labs, and we are unable to accommodate legitimate requests for secure research space by new hires because there is not enough such space available to the department.

Needed Changes in Resources

The department desperately needs access to classrooms with projectors and Internet connectivity. The College has made efforts to provide all faculty members with laptops or tablet PCs for lesson preparation and classroom presentations, but these computers need to be updated regularly. While the need for an improved educational technology infrastructure holds throughout the College, it is particularly critical for our department, which teaches *about* technology as well as *using* technology. Faculty working with obsolete equipment (often purchased with personal funding) is not satisfactory.

The Technology Fee is not handled well from our department's perspective. We understand that much of the difficulty arises from constraints that are outside the College's control, and we are aware that there are efforts underway to address some of them. But we mention here that: (1) The annual-proposal model for tech fee projects makes it difficult for us to plan intelligently. We don't know whether we will be funded for a project next year, and we certainly are unable to construct plans that carry us forward for the longer term. (2) The current guidelines prohibit funding for projects that teach about technology as opposed to projects that expand the use of technology for large populations of the student body. Not only does this policy lead to lowest-common-denominator projects that provide generic resources that don't address the needs of *any* particular discipline, it has a particularly negative effect on our department where, truly, ”the medium (the technology) is the message!”

We need a budget for maintaining our research infrastructure. We have recently hired a number of talented young researchers, and it is unrealistic to expect them to be able to get outside funding for shared resources that the College should be providing. Network upgrades, servers, and even big-ticket research equipment for such things as image and video capture need to be acquired on an ongoing basis.

Full-Time Faculty

Number

The department currently has 21 full-time faculty members, including two new hires who started in the Fall 2005 semester. Sixteen of these 21 faculty members also hold positions as members of the CUNY Ph.D. Program in Computer Science. We are currently searching for a new assistant professor, and there is also one deferred replacement line.

Ranks, tenure, and ages are distributed as shown in Table 1.

Rank	Number	Tenured		Avg Age
Full Professor	11	11	100%	54
Associate Professor	3	3	100%	47
Assistant Professor	5	1	20%	38
Lecturer	2	2	100%	44
Totals	21	17	81%	49

Table 1. Distribution by rank of full-time faculty in the Computer Science Department.

Specializations

The subdisciplines of computer science have traditionally been classified into the areas of theory, software, hardware, and methodology. An important emerging area is bioinformatics, the application of computational principles to various aspects of biological research including genome, protein, and taxonomic analyses. Table 2 lists the self-reported areas of interest, both for teaching and for research, for all the faculty in the department. Figure 1 summarizes the data. Perhaps not unexpectedly, the department's main areas of interest and expertise lie in the traditional computer science areas of software and theory.

Name	Research	Teaching
Boklan	T (Cryptography)	T (Cryptography), O (History of Science)
Brown	M (Queuing Theory)	
Chen	S (Web Programming), M (Data Mining)	S (Web Information Processing)
Fluture	B (Pattern Recognition in 3D MRI)	S (Operating Systems, Distributed Systems), T (Discrete Structures)
Ghozati	H (Networks, Parallel Processing)	H (Computer Architecture, Networking)
Goldberg	B (Biomedical Imaging), S T (Algorithms)	M (Computer Vision), T (Genetic Algorithms, Discrete Structures)
Gross	M (Computer Vision), T (Compression, Digital Geometry)	S T (Algorithms), T (Computability), O (Research Practicum)
Kong	T (Digital Topology)	S (Programming Languages, Compilers), T (Discrete Structures)
Kwok	M (Information Retrieval, Statistical Language Processing)	M (Information Retrieval, Statistical Language Processing)
Lord		S (Web and Database Programming, Assembly Language, Data Structures)
Obrenić	T (Algorithms, Graph Embeddings)	T (Discrete Structures, Theory of Computation), S (Databases)
Phillips	S T (Software Engineering, Algorithms)	M (Image Processing, Document Understanding)
Reddy	B (Protein Analysis)	B (Research Practicum)
Ryba	T (Computational Group Theory)	S (Programming), T (Discrete Structures, Computability), O (Math courses in Math Dept., Ed Dept.)
Sy	M (Bayesian Probability, Inference), H S (Wireless)	M (Data Mining), H S (VoIP)
Vickery	S (Web Design Standards), H (Logic Design Technologies)	S (Systems Programming, Web Programming, Software Design), H (Hardware, Architecture and Logic Design)
Waxman	T (Graph Algorithms), O (Computer Science Education)	S (Programming, Data Structures), S T (Algorithms)
Whitehead	T (Continuous Computational Complexity)	S (Operating Systems, Distributed Systems), T (Theory of Computation)
Xiang	M (Graphics, Image Processing)	S (Data Structures, Assembly Language), H (Computer Organization), M (Graphics)
Yukawa	S (Object Oriented Databases)	S (Object Oriented Databases, Programming Languages)
Zheng	H (Wireless and Mobile Computing, Reconfigurable Systems), M (Video compression)	H (Hardware Organization)

Table 2. Individual Faculty Members' Research and Teaching Areas of Expertise. The letters B, H, M, S, T, and O indicate the areas of Bioinformatics, Hardware, Software, Methodology, Theory, and Other.

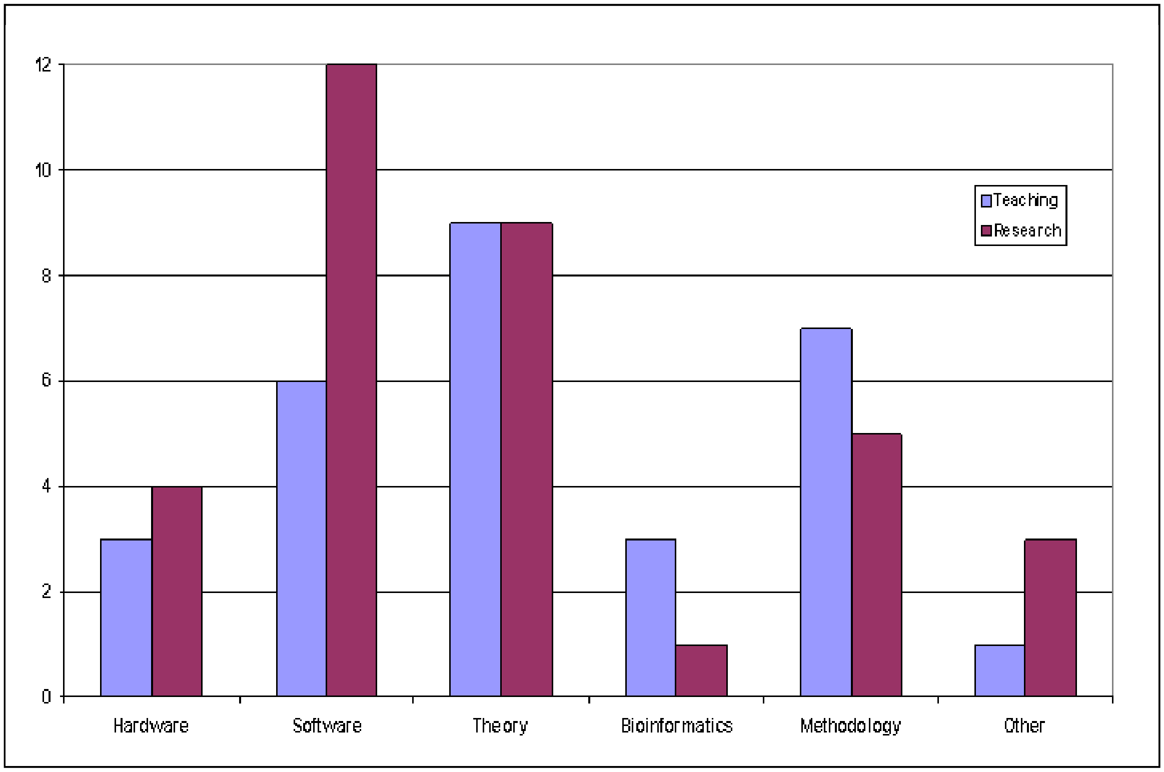


Figure 1. Total number of faculty listing teaching and research interests in the areas of Hardware, Software, Theory, Bioinformatics, Methodology, and Other areas of the Computer Science discipline spectrum.

Scholarship and Creative Activity

The department can boast of researchers who are top-ranked world-wide in the areas of group theory, digital topology, cryptanalysis, and text retrieval. Other faculty are actively productive in the areas of wireless security, vision systems, and algorithm design. We have several recent new hires who are demonstrating strong presence in several areas of research, including bioinformatics, web technology, software engineering, embedded systems, and computer architecture. Table 3 gives the number of publications in various categories for each full-time faculty member in the department, along with their internal (PSC-CUNY) and external grant activity for the past five years.

Our faculty have published in dozens of different journals (over sixty), including publications of the ACM (*Journal of the ACM, Computing Reviews*, and *Transactions on Graphics*) and the IEEE (*Transactions on Computer Vision, Transactions on Pattern Analysis, Transactions on Parallel and Distributed Systems, Transactions on Software Engineering, Transactions on Systems, Man, and Cybernetics, Computer*, and *Graphics and Applications*). Other journals include, among others: *Cryptologia, SIAM Journal of Discrete Mathematics, SIAM Journal on Computing, Mathematical Modeling and Scientific Computing*. Also, the Journals of: *Algebra, Algorithms, Complexity, Computer and Information Technology, Computer Research and Development, Computer Vision and Image Understanding, Group Theory, Interconnection Networks, Logic Programming, Symbolic Programming*,and *the American Society for Information science*, among others.

Conferences at which our faculty have presented papers in the last several years include: *ACM SIGIR Conference on Research and Development, ACM Symposium on Parallel Computation, American Mathematical Society Conference, Expert Systems In Government Symposium, IEEE International Conference on Systems, Man & Cybernetics, IEEE-CS Workshop on Computer Vision, IEEE Symposium on Parallel Computation, INFORMS, International Conference on Combinatorics, Graph Theory and Computing, International Conference on Data Analysis, International Conference on Document Analysis* *and Recognition, NADA, NADE, SIAM, SPIE, Workshop on Computer Architecture Education*, and *Workshop on Reconfigurable Computing Education*.

The vibrancy and productivity of the department also manifests itself in the activities of our faculty that fall outside the usual measures of research productivity. While the fundamental principles of computing don't change any more rapidly than the core areas of other disciplines, computing's manifestations in the real world outside of academia are evolving at an extremely rapid pace. For our department, this means that courses, even our core courses, go "stale" distressingly soon after their introduction and need constant revision on a minor or major scale in order to track the current state of the art. Much of the department's strength lies in the time and energy our faculty put into keeping their courses up to date on an ongoing basis.

Name	In Preparation	Books	Book Chapters	Journal Articles	Conference Proceedings	Other	PSC-CUNY Awards	No. External Grants	External Grant Amounts
Boklan	0	0	0	4	0	0	0	1	$50,000
Brown	0	0	0	0	0	0	0	0	$0
Chen	0	0	0	0	8	0	0	0	$0
Fluture	1	0	0	2	2	1	0	0	$0
Ghozati	0	0	0	0	0	0	0	0	$0
Goldberg	5	1	1	10	6	4	0	0	$0
Gross	0	1	0	0	1	0	0	0	$0
Kong	0	0	1	4	5	3	0	1	$100,000
Kwok	0	0	0	0	14	1	1	4	$1,200,000
Lord	0	0	0	0	0	0	0	0	$0
Obrenić	0	1	0	2	0	0	0	0	$0
Phillips	0	0	1	3	8	0	3	1	$64,000
Reddy									
Ryba	4	1	0	5	0	0	1	0	$0
Sy	0	0	3	3	5	3	0	0	$0
Vickery	0	0	0	0	2	3	4	0	$0
Waxman	2	0	0	2	3	2	0	0	$0
Whitehead	2	0	0	1	0	0	4	0	$0
Xiang	1	1	1	0	3	0	0	0	$0
Yukawa	0	0	0	0	0	0	0	0	$0
Zheng									
Sum	15	4	6	36	57	19	13	7	$1,364,000

Table 3. Number of publications, PSC-CUNY grants, and external grants for current full-time faculty for the past five years. Missing data are for new hires.

Teaching Outside Department

College

The School of Education at the College offers a program called *Time 2000* for Secondary Education Mathematics students. A. Ryba of our department teaches two courses in the Time 2000 program. One is a mathematics course, and the other is a specialized version of our introductory course geared to preparing high school teachers who will teach Advanced Placement courses in computer science.

Graduate School

One member of our department, Prof. T. Brown, is the executive officer of the Ph.D. Program in Computer Science at the CUNY Graduate Center. Prof. Brown is also the director of the CUNY Institute for Software Design and Development at the Graduate Center; he teaches one course a year at the Graduate Center, but has no teaching responsibilities within the department at this time.

Various members of the faculty teach courses at the Graduate Center, but not on a regular basis. Recent offerings by our faculty have included courses on cryptanalysis during the spring terms of 2005 and 2006 by K. Boklan, a course on Web Information Retrieval by J. Chen, and a course in Statistical Data Mining by B. Sy.

In addition, six of our faculty have mentored or are mentoring a total of nine Ph.D. students through the graduate center over the past five years. Current faculty and the number of students mentored are:

Kong:	1
Chen:	2
Goldberg:	2
Xiang:	1

Analysis

Breadth of Preparation

As indicated previously, faculty interests cover the spectrum of hardware, software, theory, bioinformatics, and a variety of applied areas that we call "methodology." Individuals' interests typically shift over time as technologies change. While some researchers do maintain continuing interest in their research areas over many years, a large portion of our faculty are constantly re-educating themselves in order to keep abreast of the times. For many, the specifics of one's preparation, except for the fundamental principles of science and technology learned, are soon nearly irrelevant because of this constant need to keep up to date.

Table 5 lists the institutions from which faculty earned their doctorate degrees, the discipline in which the degree was awarded, and the year. The department can boast that fifteen distinct and distinguished universities around the world have educated our faculty.

Name	Institution	Discipline	Year
Boklan	University of Michigan	Mathematics	1992
Brown	New York University	Operations Research	1971
Chen	Technical University of Munich	Engineering	1999
Fluture	City University of New York	Computer Science	2004
Ghozati	Columbia University	Electrical Engineering	1976
Goldberg	New York University (CIMS)	Computer Science	1989
Gross	Columbia University	Computer Science	1992
Kong	Oxford University	Computer Science	1986
Kwok	University of Manchester	Physics	1965
Lord	City University of New York	Computer Science	1995
Obrenić	University of Massachusetts at Amherst	Computer Science	1993
Phillips	University of Maryland	Computer Science	1984
Reddy	University of Hyderabad	Life Sciences	1988
Ryba	University of Cambridge	Mathematics	1985

Vickery	City University of New York	Experimental Psychology	1971
Waxman	New York University	Computer Science	1973
Whitehead	University of Warwick	Mathematics	1975
Xiang	University of Buffalo	Computer Science	1988
Yukawa	Waterloo University	Computer Science	1987
Zheng	University of Nevada	Computer Engineering	2005

Table 5. Institutions from which full-time faculty received their doctoral degrees, plus the disciplines and years in which they were awarded.

Affirmative Action Goals

The faculty ethnic distribution is 55% white and 45% Asian. There are no Blacks, Hispanics, or Native Americans represented on the faculty.

Only 20% of our full-time faculty members are female, but this low proportion is in fact considerably higher than the national average for computer science programs. According to current data from the Computing Research Association, women accounted for 18% of newly hired, tenure-track lines, 16% of assistant professors, 12% of associate professors, and just 10% of full professors for the 2004-05 academic year.

While the department has no obvious biases in the makeup of its faculty, especially given the ethnic and gender mix of our students, we actively look for opportunities to increase our diversity while working within the equal opportunity guidelines of the College.

Age Distribution

As Table 1 shows, the average faculty age for the department is 49, with the expected increase in average age with increasing tenure-track ranks. There seems to be no issue of concern with regard to age.

The percentage of our faculty with tenure (80%) is higher than the College at large (75% according to the College's web site) and is very high compared to the national average for public research and doctoral institutions (about 45% in 1998-99 according to the NEA Research Center. The department is "top heavy" partly because we we unable to hire any new faculty members during the period from 1994 to 1998.

College Service

The department is particularly well-represented on the various committees of the Academic Senate. Dr. K. Lord is chair of the Undergraduate Curriculum Committee and serves as Assistant to the Provost for Instructional Technology; in this latter position, Dr. Lord also serves as an ex-officio member of the Technology and Library Committee. Prof. C. Vickery is chair of the Nominating Committee and is a member of the Technology and Library Committee. Prof. Z. Xiang is a member of the Graduate Scholastic Standards Committee.

Recent Recruitment

We have hired four new faculty in the past three years. Each of them brings an exciting dimension to the department in his own way. J. Chen has a strong background in web technologies (with experience at Microsoft and elsewhere) as well as embedded systems design. K. Boklan, with experience at the NSA, is an international authority on cryptanalysis. Our newest hires, B. Reddy and J. Zheng, are extremely strong in the areas of bioinformatics and embedded systems design.

Altogether there have been eight new-hires since 1994. However, five faculty positions in our department have been vacated during the same period, and one member of our faculty, Prof. Brown, is now based at the Graduate Center (he is the Executive Officer of the Ph.D. Program in Computer Science). So the effective size of our faculty is still much the same today as it was 12 years ago.

Search Process

Once the department has been authorized to search for a new hire, the department's Personnel and Budget Committee follows the standard college procedures required by the College's Affirmative Action Officer. We advertise the position and the areas of expertise in which we are interested in appropriate online venues (such as the ACM web site), we receive resumes and letters of recommendations, we draw up a matrix of candidates and their qualifications, and place candidates into an unranked top tier, a rank-ordered list of other qualified candidates, and an unranked set of unqualified candidates. Once the Affirmative Action Officer has certified that our candidates have been categorized objectively, we invite all candidates from the top tier in for interviews and presentations of their research. After these interviews we decide which of the top-tier candidates we would like to hire and make offers to them one at a time until we either get an acceptance or receive rejections from all top-tier candidates in whom we are interested. If we exhaust our top-tier pool, we invite candidates in from the second-tier one at a time until we either make a successful offer or declare the search a failure.

Selection Criteria

The notion of "selection criteria" for our searches is deeply colored by the salary scale we are able to offer prospective candidates. The Computer Research Association publishes an annual survey of faculty salaries in Computer Science, called the "Taulbee Survey" in honor of the man who first conducted the survey in the 1970's and 80's.

The Taulbee Survey differentiates among institutions by the "academic ranking" of their computer science departments. The main research institutions like Stanford and MIT are in the first rank (top 12 institutions). Columbia and NYU are in the second rank (next 12), SUNY Stony Brook is in the third rank, and the CUNY Graduate Center is included in the set of unranked (below 36) respondents to the survey. As one might expect, salaries at higher ranked institutions are greater than salaries at lesser-ranked institutions. That is, CUNY salaries are not competitive with highly-ranked schools within the New York Metropolitan region such as NYU and Columbia, and they are not even comparable to other public institutions in the area such as Stony Brook and Rutgers, which are in the third tier.

So, how do CUNY salaries compare to salaries at other schools that are ranked below 36 or are unranked? Well, the *maximum* CUNY Assistant Professor's salary is $65,388. And the average *minimum* computer science assistant professor's salary among institutions that are ranked below 36 or are unranked, according to the most recent Taulbee Survey [Zweben 2005], is $72,691. That's an 11% salary disadvantage to start with. When this salary disadvantage is coupled with the high cost of living in New York City and the relatively heavy teaching load required of CUNY faculty, we consider ourselves very fortunate to have attracted the dedicated and highly-qualified group of faculty we do have. But the fact remains that those people with the best research potential, even those who want to be in the New York area, will typically look elsewhere before coming to CUNY.

The department has been able to recruit highly-qualified faculty despite our poor competitive position. In part this is because recent declines in CS enrollments nationally have made jobs scarce at a time when we have been fortunate enough to be hiring. However, this situation is only temporary, and we anticipate greater and greater difficulties in attracting good people into our program as the economy's tech sector rebounds.

Evaluation

Scholarly and Creative Activity

One way to improve levels of scholarly activity is to reduce teaching loads so that people have time to do research. To this end, it is encouraging that we have been able to offer our most recent hires startup packages that give them the chance to establish levels of outside funding that might allow them to "buy out" some of their teaching responsibilities. The flaw in this scheme, however, is that the department does not have a body of senior faculty with research programs in place that could provide the necessary support system for helping these junior faculty get established before their startup benefits give way to the teaching load grind imposed on the rest of the members of the department, college, and university.

One way this situation can be turned around is for the department to hire senior people with established research programs. Given proper funding incentives, the department should be able to recruit good people who are keenly interested in relocating to the New York metropolitan region. We regularly receive applications from such people when we advertise our searches, but we have not been authorized to hire such much-needed talent in the past. Without infusion at the top, the department will remain primarily a teaching body rather than a strong center of scholarship.

Teaching

Teaching is our strength but, as indicated above, it drains our scholarly resources. Table 6 shows the teaching loads for each department in the Division of Mathematics and Natural Sciences. As the columns labeled "Load" indicate, our department has the highest teaching load in the division at the undergraduate level.

Department	Undergraduate					Master's					Doctoral	
	Number of FTE Faculty	Load (hrs/wk)	FT Hrs	PT Hrs	% PT	Number of FTE Faculty	Load (hrs/wk)	FT Hrs	PT Hrs	% PT	Number of FTE Faculty	Load (hrs/wk)
Biology	19.4	10.8	108.0	127.3	54.1	4.1	8.2	26.9	11.0	29.0	1.7	6.0
Chemistry	18.5	8.6	42.8	183.1	81.1	1.7	10.5	18.2	0.0	0.0	1.4	9.0
Computer Science	**20.8**	**11.4**	**126.0**	**131.5**	**51.1**	**8.7**	**8.6**	**67.2**	**12.0**	**15.2**	**1.7**	**5.6**
EES	9.7	9.4	62.3	38.0	37.9	4.9	5.1	22.9	6.0	20.8	0.6	8.6
Mathematics	37.1	9.0	208.4	185.5	47.1	4.9	7.6	37.0	0.0	0.0	2.1	1.7
FNES	26.2	8.5	74.0	230.6	75.7	6.3	9.3	45.6	19.0	29.4		
Physics	15.4	7.5	57.6	97.0	62.7	0.9	9.4	3.0	8.0	72.7	0.4	5.9
Psychology	23.6	10.4	144.9	129.0	47.1	6.3	10.6	61.3	3.0	4.7	6.5	9.5
Division Average	**21.3**	**9.5**	**103.0**	**140.3**	**57.7**	**4.7**	**8.7**	**35.3**	**7.4**	**17.3**	**2.1**	**6.6**
College Sum/Avg	556.7	9.1	2896.9	3089.4	51.6	191.5	8.0	1110.9	654.7	37.1	49.6	7.0

Table 6. Teaching loads in the Queens College Division of Mathematics and Natural Sciences. Data are taken from the 2004-05 edition of the Queens College Fact Book [QC 2005], "Teaching Load Analysis - Derived Data," pages 198-208.

Over half of the teaching in our department is performed by adjuncts. As Table 6 shows, our department is about average compared with the rest of our division in this regard. However, were we to attempt to obtain accreditation for our program, *this fact would disqualify us*. The Computing Accreditation Commission (ABET) specifically requires: [ABET 2005, page 2]

> III-2. Full-time faculty members must oversee all course work.
> III-3. Full-time faculty members must cover most of the total classroom instruction.

With graduate student adjuncts carrying full responsibility for many of the courses, it is very difficult to assure high levels of expertise in the classroom. And with full-time faculty stretched thin covering their own heavy course loads, there is little "bandwidth" available for monitoring and mentoring adjuncts to be sure they deliver the best quality instruction possible. Furthermore, this use of teaching assistantships as a way of funding the equivalent of adjunct teaching staff means there are no teaching assistants in the traditional sense to help full-time faculty meet the demands of their own courses.

Service

As pointed out in the analysis of departmental service, our department is a particularly strong contributor to various college committees and administrative functions.

Faculty Development

Faculty development is a critical issue for the Computer Science Department. We reiterate here the point that the fundamental principles of computing change no more rapidly than the fundamental principles of other disciplines, but the manifestations of these principles change much more quickly. If we were to present fundamental principles to our students in the context of yesterday's technology we would appear to be teaching irrelevancies instead of the unchanging truths that they are. To track today's technologies requires constant monitoring of current trends and constant development of new technological skills on the part of faculty.

Just as our teaching cannot be allowed to stagnate, so too with our research efforts. In this regard we are not so different from other disciplines in the sciences. Arguably, computer science is subject to shorter term shifts in what the "hot" research topics are than other disciplines. Regardless, keeping up with current trends in one's research area requires a tremendous amount of effort. What seems to be special to computer science compared to other disciplines, however, is that faculty typically change the entire focus of their research more often than other scientists. We don't have hard data to support this assertion, but within our department we have seen numerous examples of faculty members who were hired when they worked in one area and who had switched to another one even before they were awarded tenure.

Aside from self-study, the major vehicle for faculty development is participation in events ranging from traditional workshops and conferences to more prosaic trade shows and vendor presentations. Aside from the time commitment required, there is a need for travel funds to support these efforts. One of our needs is to increase the availability of these funds to allow our faculty to take greater advantage of the opportunities available for attending a range of such events on an ongoing basis.

Staff Development

As an aside to the issue of faculty development, we would like to point out that our department also needs to address the issue of staff development as well. Maintaining the research and teaching infrastructure of a computer science department requires staff who are able to go beyond conventional IT skills to make sure that our networks, software, and computing platforms are all maintained, planned, and coordinated to meet the demands of faculty members who are intimately involved with various aspects of computing technology that extend far beyond the normal realm of a typical organization's information infrastructure.

We have been fortunate in being able to hire two individuals who provide excellent support for the unique nature of our infrastructure. But like any IT professionals, they need access to regular training to keep their skills in tune with current technologies. Furthermore, such training is perceived as in important job benefit that is necessary for keeping high-quality staff on board. Released time and funding for these individuals to attend workshops, training sessions, and vendor presentations are critical for the vitality of the department and must be fully funded.

Research and Funding

We are fortunate to have several world-class researchers in our department. Our most productive member in terms of outside funding is K. L. Kwok, whose work on natural language information retrieval has obtained over a million dollars in DARPA funding over the past five years. Several junior faculty members are currently bringing their research programs up to speed and actively engaged in efforts to attract outside funding. And a number of our faculty regularly take advantage of the resources available through the Professional Staff Congress - City University of New York (PSC-CUNY) Research Award Program.

The University Committee on Research Awards, a faculty committee, distributes the PSC-CUNY Research Award Program awards, and the Research Foundation of CUNY administers them. Preference is given to junior faculty in the allocation of funds, which generally amount to a few thousand dollars a year per awardee.

Adjunct Faculty

Recruitment

We draw our adjunct faculty from two pools: CUNY graduate students, and professionals drawn from local industry. There is very little in the way of recruitment in the case of graduate students: they generally come to us as part of the process of being students. In some cases, adjuncts who began as graduate students have stayed on as adjuncts even after completing their graduate studies.

So far, we have not actively recruited adjuncts from industry. We have one adjunct line that is paid for by Computer Associates that covers a CA employee who teaches for us. Otherwise, there are only two adjuncts who come to us from industry and one whose primary affiliation is another college. They have been teaching for us for a number of years.

Selection

The selection process for adjuncts occurs infrequently enough that it is not formalized. If we need someone to cover a course and can't find a graduate student who is qualified and capable of managing it, the chair engages in an informal process of evaluating unsolicited resumes, recommendations from colleagues, and similar approaches to try to locate a suitable individual who matches our needs.

Supervision and Development

We generally treat our adjuncts as responsible and self-sufficient individuals. We maintain syllabi of all courses that we offer and and make sure that new adjuncts receive copies of the appropriate material and discuss the nature of the course with one of the full-time faculty members who is most familiar with the course. By making a full-time faculty member the adjunct's point of contact for curricular matters we insure that the adjunct will not be left to flounder.

Evaluation

Adjuncts are included in the course evaluation process based on a questionnaire distributed by the College. The Personnel and Budget committee reviews the results of these questionnaires, as well as unsolicited feedback we regularly receive from students, in determining whether to reappoint adjunct faculty to their positions. Over the years we have established a "working set" of adjuncts who are a valuable resource for the department.

Curriculum and Enrollments

Enrollments

Following the "Dot.com Bust" of 2000-01, computer science enrollments declined nationwide, and at Queens as well. Nationally, "the number of newly declared CS majors has declined for the past four years and is now 39 percent lower than in the Fall of 2000. Enrollments have declined 7 percent in each of the past two years." [Vegso 2005] At Queens, from 1995 to the peak of the dot-com boom in 2000, the department's number of annual FTE students increased 78%, while from 2001 to 2005 the number decreased 50%. More recently, the total number of students in all sections of our undergraduate courses decreased 17 per cent for the Spring 2006 semester compared to a year earlier. At the same time there was an enrollment drop of 4 per cent in the enrollments in our Master's courses for those two semesters.

Although we think of ourselves as a normal department within the College's liberal arts curriculum, we must admit that students tend to view us as "the place to go if you want a job in technology," and our enrollments are subject to the vagaries of the economy and perceived availability of jobs in technology.

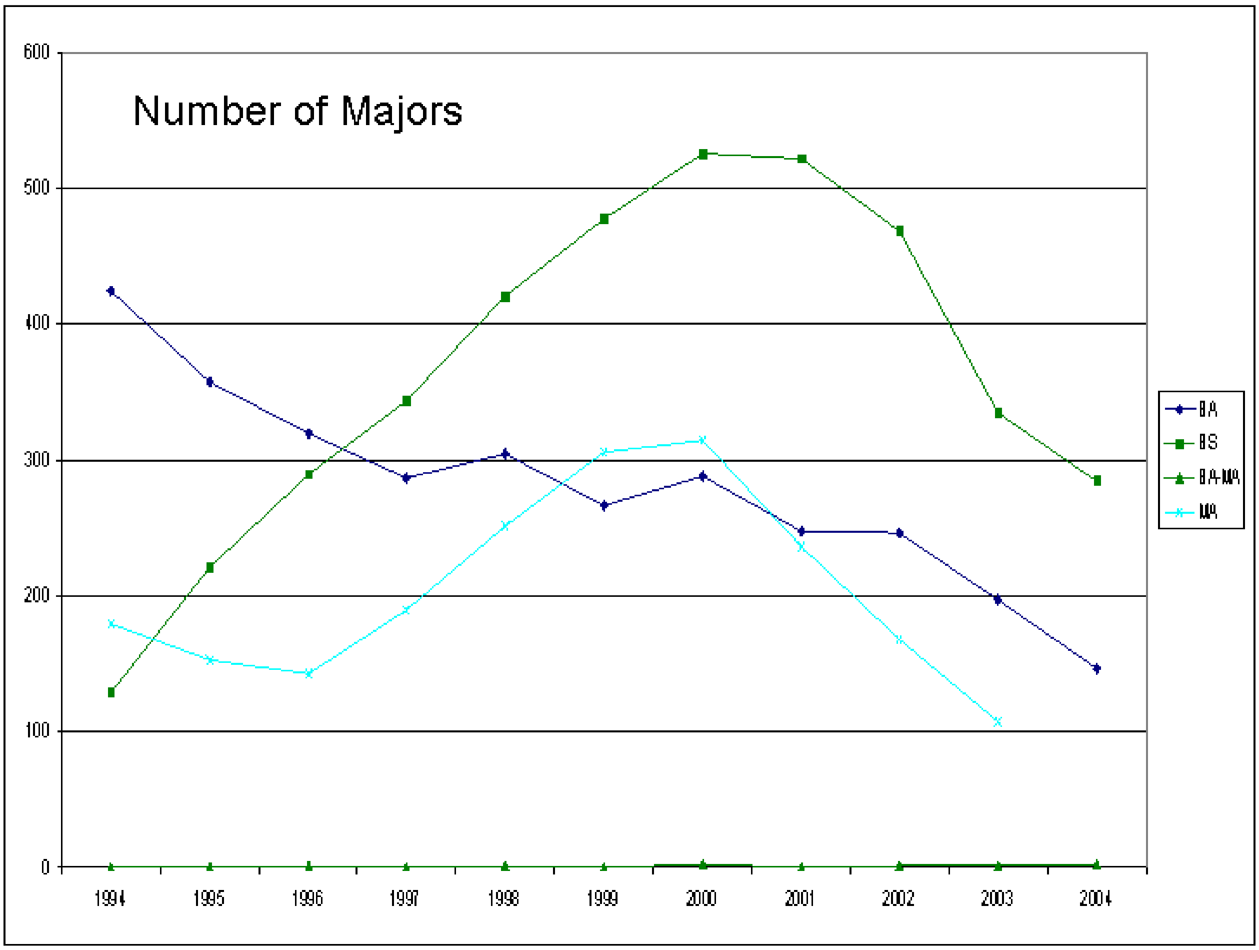


Figure 2. Number of Queens College Computer Science Majors: 1994-2004. Data are for students registered for the BS and BA degrees, the combined BA-MA degree, and the MA degree.

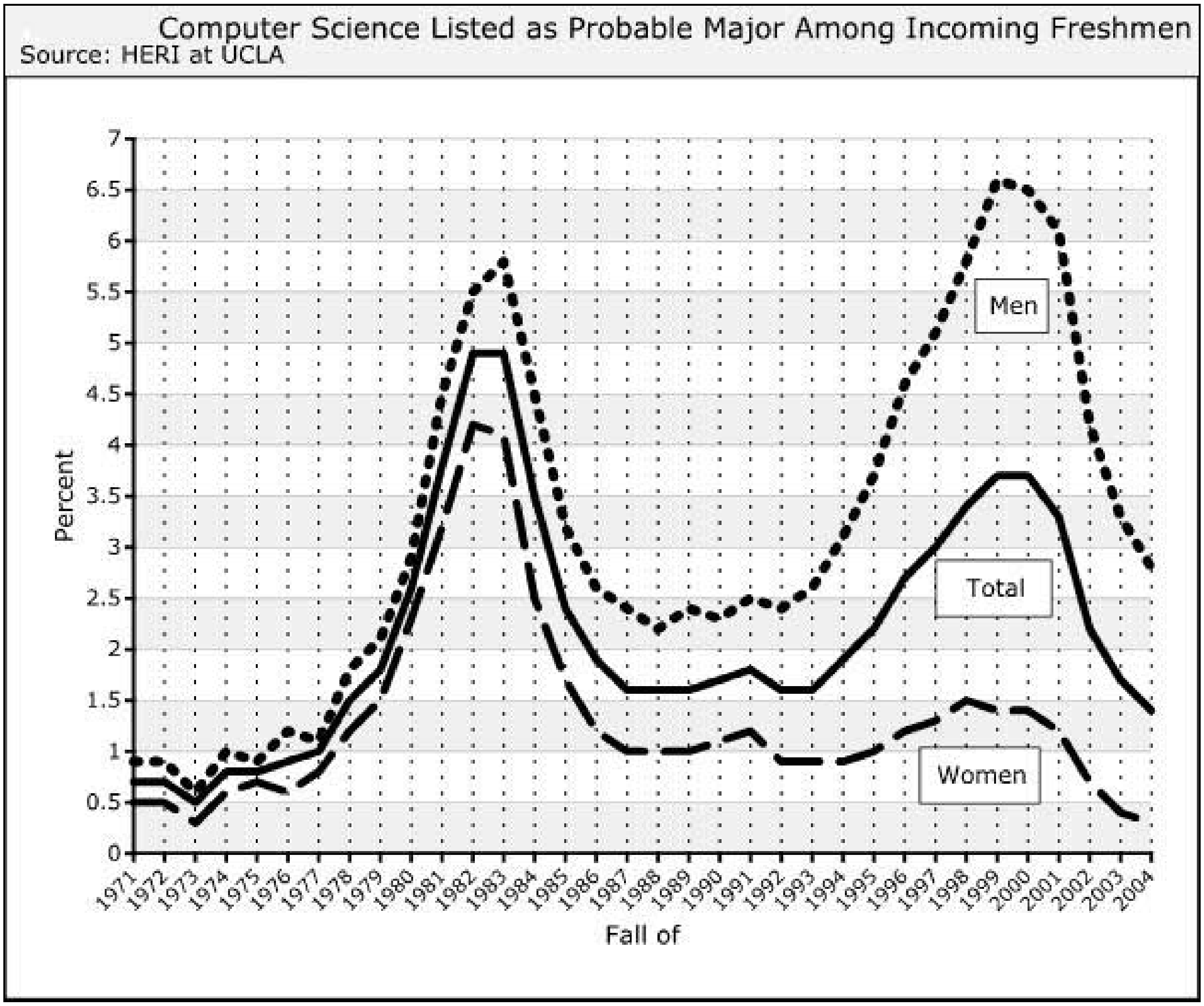


Figure 3. National data for number of Computer Science majors since 1971, taken from [Vegso 2005].

The department offers two undergraduate majors, one leading to the BA degree and another leading to the BS. These are described more fully in the Programs for Majors section below. We also offer a combined BA-MA degree option, as well as a separate MA degree. Figure 2 shows the enrollment trends for all of these degrees over the past decade. What appears to be most interesting about this data is the difference between the trends for BA and BS degree enrollments. BA enrollments have been declining at a smooth rate of about 5% a year over the period shown, while the number of BS majors closely parallels the national trend shown in the right-hand part of Figure 3. However, the explanation for the difference between the BA and BS trends is simply that the BS option is a recent addition to our curriculum, having been offered for the first time in 1993, just one year before the data presented in Figure 2 begin. Students generally prefer the BS option, and the left side of the figure reflects the ramping up of the new degree option at the expense of the BA.

Figure 4 shows the number of bachelor's and master's degrees in Computer Science awarded over the past ten years. These data lag the current number of majors and instead reflect the state of the "degree pipeline" rather than current enrollments. As such, the trends for the bachelor's degrees are still continuing upward, whereas the master's degree function has already peaked, reflecting the shorter pipeline for the 1-2 year master's pipeline compared to the 4+ year pipeline for the bachelor's degrees. Most importantly, however, the data suggest that our students are completing their degree requirements in a timely fashion.

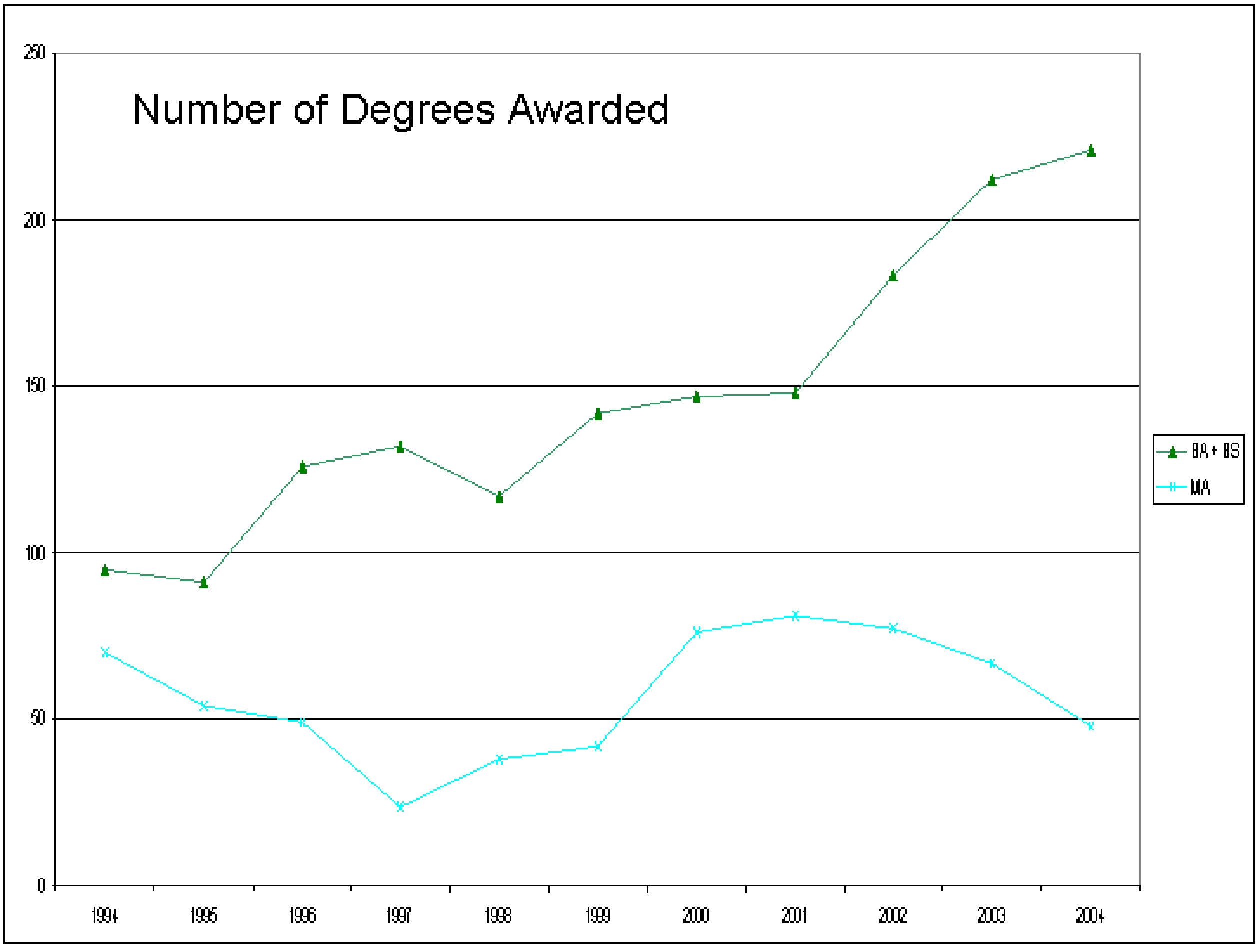


Figure 4. Number of CS degrees awarded: 1994-2004. Data are for the sum of BA and BS degrees, and the MA degree.

Minors

We present the data for the number of Computer Science minors in Figure 5. There is no clear trend over the time span shown. Given the small number of students who are registered for the program, we feel that there is little significance to the variations that are seen there.

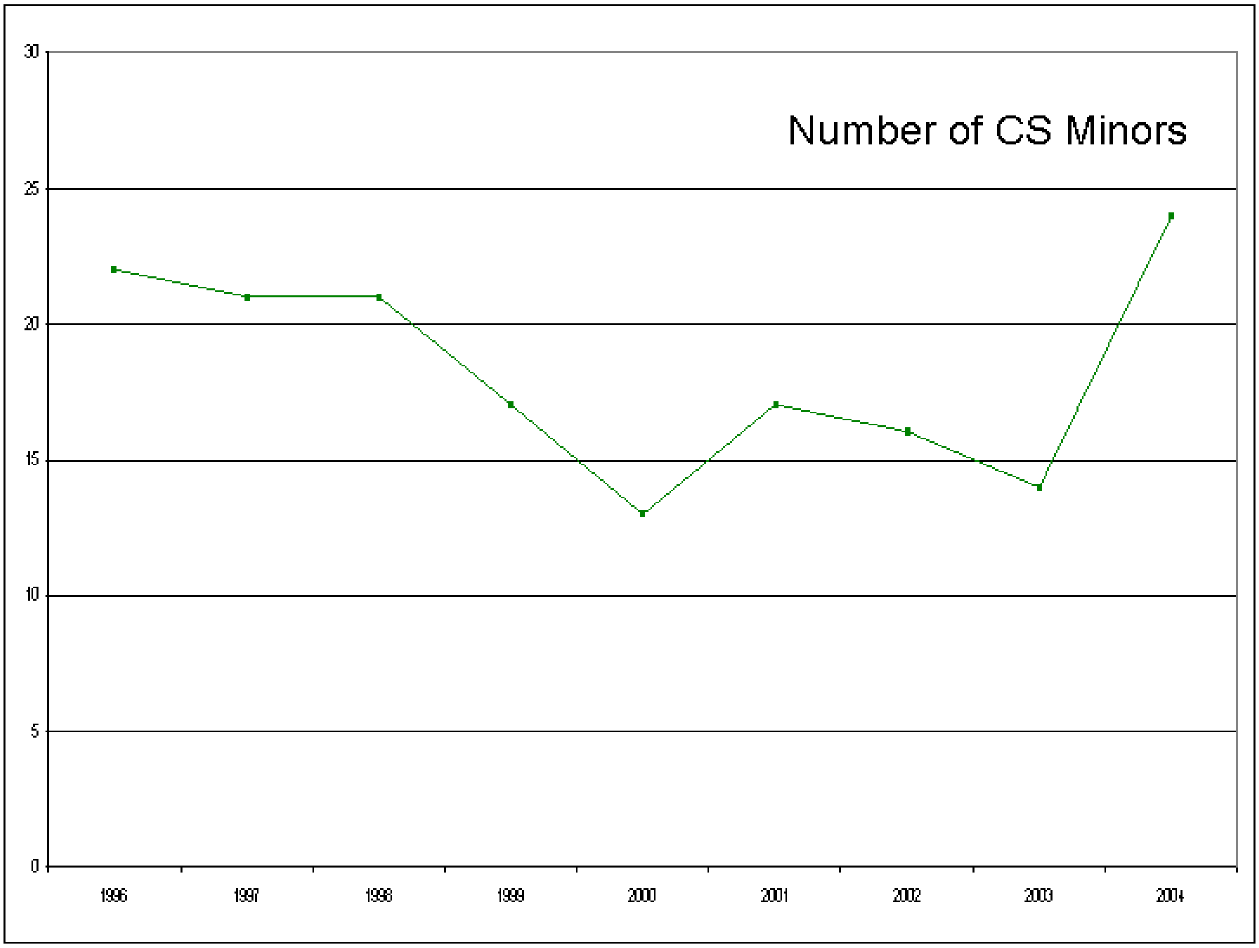


Figure 5. Number of CS Minors: 1996-2004. Data for our new CIT minor are not yet available.

In addition to the CS minor, which is a proper subset of the BA major, the department has a new "Computer Information Technology" (CIT) minor that was fully implemented for the first time during the 2005 academic year. Although there is no data available for this program yet, we think it holds considerable promise for attracting more students into the department. The CIT is described more fully in the Programs for Non-Majors section below.

FTEs

Figure 6 presents the number of Full-Time Equivalent (FTE) students served by the department since 1984. Comparing that figure with national data for the comparable time interval in Figure 3 shows that our department's number of FTE's has closely paralleled the national trend for number of CS majors as well as our own number of majors.

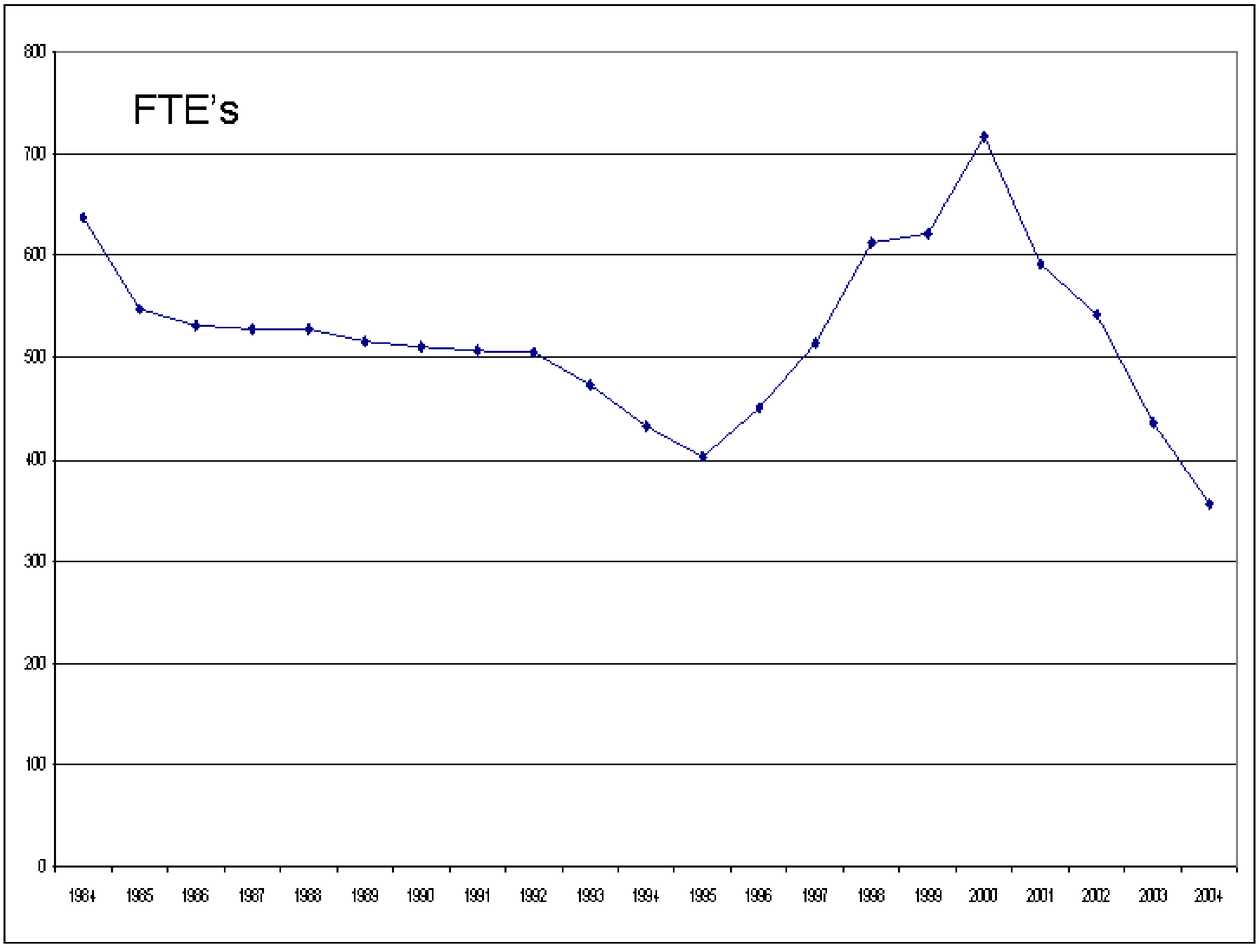


Figure 6. Number of CS FTE's: 1984-2004. Note the parallel to the total number of students majoring in Computer Science in the United States for the same time period, shown in Figure 3.

The latest figures available put the number of FTE students in the department (Bachelor's and Master's combined) at 357. This ranks us fourth in our division, with Psychology, Mathematics, and FNES ahead of us and four other departments behind us.

Curriculum: Contributions of Department to College Programs

General Education

Division: Currently we are contributing one course to the Division of Mathematics and Natural Sciences, a History of Science course being taught by K. Boklan.

College: A. Ryba of our department currently teaches a section of our introductory course especially structured for the mathematics students in the School of Education's Time 2000 program.

Computer Science 12 is our main service course for the college. Each semester hundreds of students take this course to learn fundamental concepts in various aspects of computing, including instruction in basic computing skills from word processing to basic web design. A critical feature of the design of this course is that we present all material in the context of accurate models of computation. Rather than just rote rules about how to perform tasks, we stress how to reason about getting tasks done using available computing resources.

LASAR

All courses in the Computer Science majors satisfy the "Scientific Methodology and Quantitative Reasoning" LASAR requirement for the BA and BS degrees. With the ongoing revision of the General Education requirements currently being undertaking at the College, we hope to integrate our department's offerings with a more up to date view of the role of technology as one of the core elements of a liberal education today.

Evening Division

It has been a long-standing policy of the department to accommodate the needs of the many students who have work commitments that make it impossible for them to attend classes during the day. We offer evening sections of all courses needed to complete both of our undergraduate majors. Although evening sections are more often taught by adjuncts than day sections, many evening sections are taught by full-time faculty.

Weekend College

For the past three semesters, between 55 and 60 students have taken the Weekend College version of CS-12, our service course for non-majors. The course has been taught by an adjunct each time it has been offered.

Summer Session

The department offers most of the core courses within our majors during the summer, and typically includes several electives as well. The teaching is about evenly divided between adjuncts and full-time faculty.

Interdisciplinary Offerings: As science and technology have become core elements of the fabric of society, it has become essential to integrate an understanding of them in the education of tomorrow's leaders. Although there is an important role to be played by the pure computer science major, it is also critical now more than ever to integrate the department's perspective on all things technical with the broader educational process at large.

In this context, we are excited at developments that are unfolding in cooperation with the Biology Department. Several of our faculty, including M. Song (who recently left the department) and R. Goldberg have been engaged in joint research with Biology faculty and one of our most recent hires, B. Reddy, is actively involved in developing research-oriented courses that focus on issues in Bioinformatics. We feel that courses like these, as well as the ones we tailor to the needs of other programs on campus, such as the Time 2000 course mentioned earlier, are important avenues for integrating computing principles into the liberal arts curriculum at the College.

MA/PhD: The department offers its own Master of Arts degree in Computer Science, described in the Master's Degree Section. We do not offer any courses as part of any other Master's program at the College.

Over three-quarters of our faculty are members of the Graduate Faculty of the CUNY Ph.D. Program in Computer Science, headquartered at the Graduate Center. However we do not offer any 800-level courses at Queens at this time.

Graduate School

Academic, Administrative, and Financial Roles: Queens CS faculty teach some courses at the Graduate Center, as mentioned elsewhere in this document. In addition, also mentioned elsewhere in this document, the Executive Officer of the Ph.D. Program in Computer Science at the Graduate Center is T. Brown of our department. Dr. Brown also serves as the director of the CUNY Institute for Software Design and Development, a position he holds independently of his position as EO of the Ph.D. program.

Curriculum: Programs for Majors

Outline

We provide our diverse student body with a two-layered curriculum for the CS major (BA or BS), which consists of a primary core and a flexible extension. The primary core focuses on the fundamentals of hardware (computer organization, assembly programming, computer architecture), software (OOP, data structures, databases, principles of programming languages, software engineering, OS), and theory (discrete structures, design and analysis of algorithms, theory of computation). These topics ensure a solid and long-lasting foundation in the scientific principles of computing both for graduate study in CS and for a career as a computing professional.

This primary core is augmented with elective courses that extend into numerous areas of application as well as frontiers of active research and development, including bioinformatics, cryptography, data mining, internet and web technologies.

Additional enhancements come from the math (calculus, probability and statistics) and lab-based science requirements. Students are also encouraged to supplement their classroom training with internships in a variety of organizations, ranging from software companies (e.g., CA Inc.) to financial institutions to city/state agencies.

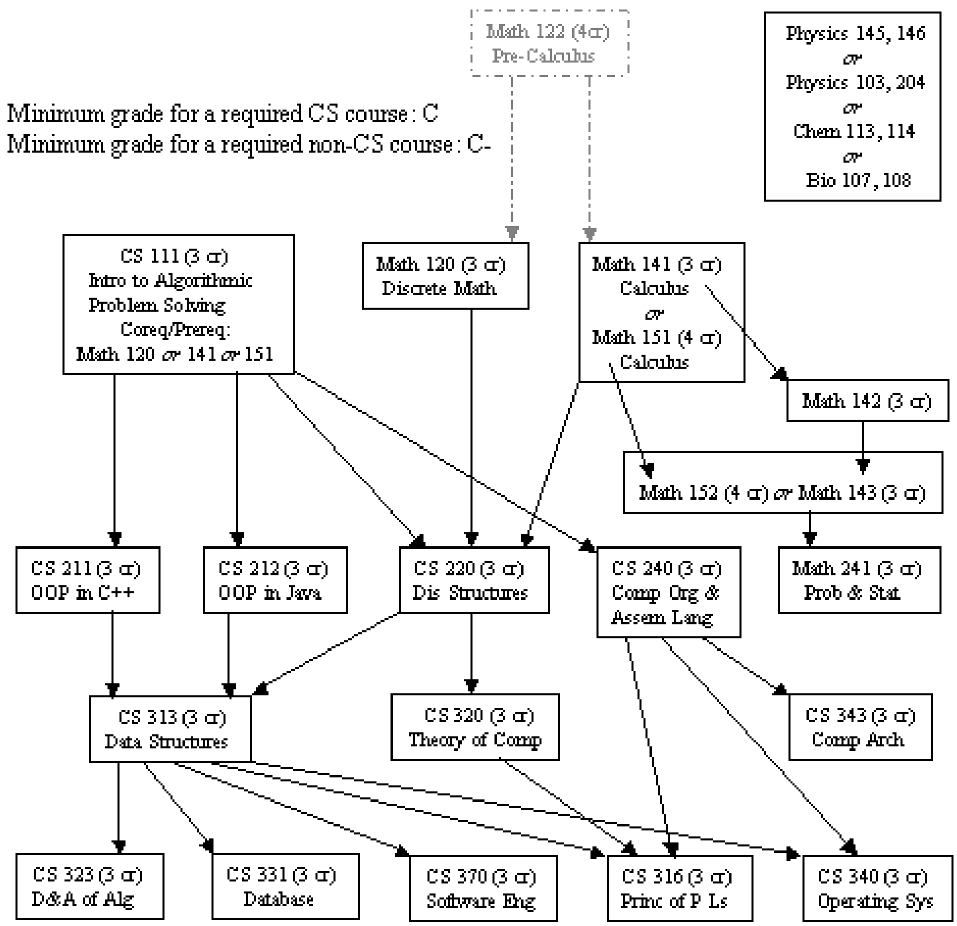
With the permission of the department, students may work on a research project and receive elective credits. Outstanding students can take Honors Readings, Honors Problems, or Honors Thesis as an elective.

The two degree options, BA and BS, differ in that the BS also requires linear algebra, has a more demanding lab sequence, and calls for five CS electives instead of three.

Typical Student Paths Through Major

We offer two undergraduate majors in computer science, one leading to the Bachelor of Arts degree and the other to the Bachelor of Science. Figure 7 shows the prerequisite structure for the BA degree program, and Figure 8 shows the prerequisite structure for the BS degree.

Computer Science Departmental Requirements —— BA
and **Prerequisite Structure** (Curriculum 2005)



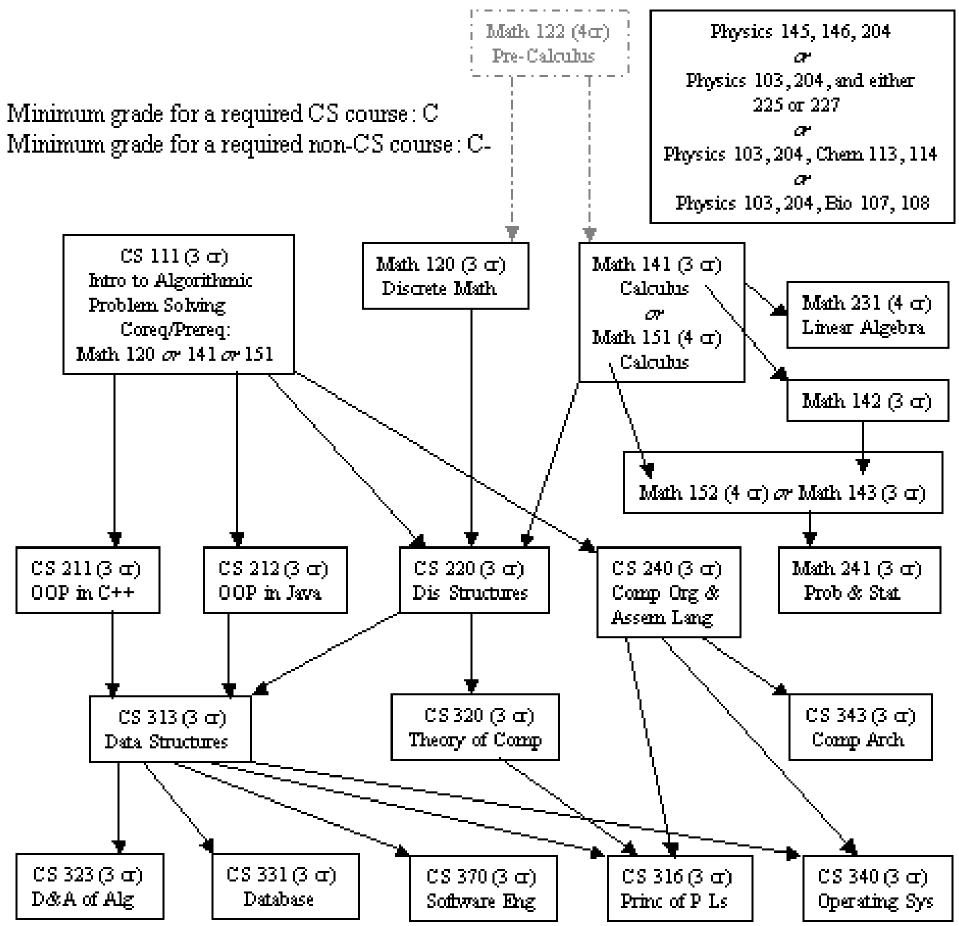
Three additional electives chosen from: (consult QC Bulletin and Class Schedule for details)

Artificial Intelligence	Data Communications	Data Mining & Warehousing
Compilers	Distributed Systems	Information Organization & Retrieval
Computer Graphics	Logic Design Lab	Internet & Web Technologies
Cryptography	Numerical Methods	Object-Oriented Databases
Operating System Programming		Special Topics in Computer Science

One approved course from Biology, Mathematics, or Physics may be used.

Figure 7. Prerequisite Structure for the BA degree. The diagram includes all required courses.

Computer Science Departmental Requirements — BS and Prerequisite Structure (Curriculum 2005)



Five additional electives chosen from: (consult QC Bulletin and Class Schedule for details)

Artificial Intelligence	Data Communications	Data Mining & Warehousing
Compilers	Distributed Systems	Information Organization & Retrieval
Computer Graphics	Logic Design Lab	Internet & Web Technologies
Cryptography	Numerical Methods	Object-Oriented Databases
Operating System Programming		Special Topics in Computer Science

One approved course from Biology, Mathematics, or Physics may be used.

Figure 8. Prerequisite Structure for the BS degree. The diagram includes all required courses.

Students in both majors generally start with CS111 and a required math course, followed by 200-level, 300-level core, and 300-level elective courses. They also take the lab-based science courses and the College-required liberal arts courses along the way.

Transfer students from community colleges typically have CS111, one or two math courses, and two or three of the required 200-level courses. They simply start with what they don't have at the 200-level and proceed from there.

Analysis of Major

LASAR: Our students receive rigorous training in computer science and information technology along with a well-rounded liberal arts education. Every CS course satisfies the Scientific Methodology and Quantitative Reasoning area requirement of LASAR.

General Degree Requirements: Students who take internships to gain real-world experience receive general elective credits towards their degree.

Advanced and Integrated Study: CS students may elect to double major, with the most likely choices being math+CS and accounting+CS. Courses required by both majors can then be applied towards both sets of requirements.

Diversity of Talents/Ways of Learning: Each semester we strive to offer a rich set of electives that cater to the diverse interests/talents of our students and reflect the dynamic and multi-faceted nature of the discipline. For example, in Spring 2006 we have: Bioinformatics, Cryptography II, Data Communications, Data Mining and Warehousing, Distributed Systems, Information Organization and Retrieval, Internet and Web Technologies, Internet Security, Mobile Computing, Next Generation Network Services, Numerical Methods, Object Oriented Databases, and Voice Over IP/WLAN.

Quality of Advisement: We have several designated undergraduate advisers with office hours both during the day and in the evening. We also have an assistant chair for undergraduate studies who is the primary contact person for such matters as transfer credit evaluation and graduation approval. The assistant chair is a member of the departmental curriculum committee, which is headed by the department's deputy chair. This enables any special needs to be addressed consistently and expeditiously by people who are responsible for curriculum issues. The students may also drop by the department office to seek help from the office staff (general information, registration, etc.), and to meet the department chair whenever necessary.

In order to somewhat alleviate the problem of having no teaching assistants to coach students who encounter technical difficulties in their study, we solicit students in good academic standing to serve as tutors. They are paid ~$10/hr by the department.

Student/Faculty Contact Outside Class: All full-time faculty are available to meet students during published office hours and/or by appointment. In addition, students can send inquiries by email at any time, and talk to professors before/after classes.

The department participates in all College-sponsored advising activities including open house, transfer credit evaluation, and sophomore day.

Relationship between Major and Specialized or Pre-Professional Programs: Students with a GPA of at least 3.5 over 15 or more credits in courses required for the major may apply for the accelerated BA/MA program, which permits them to use up to four of the required graduate core courses towards both the BA and the MA degrees in CS. Students in the BA/MA program pay the undergraduate tuition for the graduate courses until they complete a total of 120 credits. As a result, rather than taking 150 credits for both degrees (120 for the BA and 30 for the MA) students may earn both degrees with a minimum of 138 credits.

Courses Appended

There are three sources of information about the courses offered by the department:

1. The official course listings are in the College Bulletin, which is available online at www.qc.edu/college_bulletins, and reproduced in Appendix A of this document.
2. The department keeps a list of courses, degree requirements, special topics course descriptions, and other information on its website at www.cs.qc.edu
3. Most faculty maintain their own web sites for the courses they teach. Links to these individual pages are kept on the department's web site.

Comparison with Other Institutions

Table 7 compares the BS degree requirements of our department with three comparable public institutions and one private institution in the New York metropolitan region.

Inside CUNY: Brooklyn College and City College are very similar to QC in terms of institutional organization, general student body, and overall academic reputation.

Our BS program is more demanding and rigorous than that of Brooklyn College. This is evidenced by our requiring OOP in two widely-used languages, algorithm analysis plus theory of computation, more CS electives, and the lab science sequence.

On the other hand, our program is very similar to that of City College, with a bit more flexibility given to our students as demonstrated by a smaller set of required software courses and a larger set of electives.

Outside CUNY: Polytechnic University is a regional private institution in Brooklyn. Comparing the requirements in software, hardware, theory, CS electives, and lab sciences it is evident that our BS program is stronger than theirs. From their catalog course descriptions, it appears that they generally award 4 credits for courses with equivalent content to 3-credit courses at Queens, suggesting that their total number of required credits may not be directly comparable with ours.

The CS program at SUNY/Stony Brook enjoys a fine national reputation. Their BS has a stronger math component than ours. They also afford their students more freedom in choosing core courses; in comparison our core requirements are more rigidly structured. However, our approach ensures that students coming into a higher-level course are all similarly prepared academically.

	Queens	Brooklyn	City	Polytechnic	Stony Brook
Total Credits CS + Math + Science	81	52-55	84	84	80
Calculus I & II	yes	yes	yes	yes	yes
Calculus III or higher	no	no	yes	yes	Finite Mathematical Structures
Linear Algebra	yes	yes	yes	yes	yes
Probability & Statistics	yes	yes	no	no	yes
Required software courses	OOP in C++ OOP in Java Data Structures Prog. Languages Software Engineering Operating Systems Database	Programming in C Data Structures Prog. Languages Operating Systems Project	Intro to Computing Data Structures Prog. Languages Software Engineering Operating Systems Simulation Software Lab Design Project	OOP Data Structures Prog. Languages Software Engineering Operating Systems Project	OOP Data Structures Software Engineering Three from: Prog. Languages or Compiler Operating Systems Database Graphics or User Interface
Required hardware courses	Comp. Organization Computer Architecture	Assembly Language Comp. Organization	Comp. & Assembly Comp. Organization Comp. Systems Lab	Arch. & Organization	Comp. Organization One from: Networks Architecture Communications
Required theory courses	Discrete Math Design/Analysis Algo. Theory of Comp.	Discrete Math Algorithms or Theory	Discrete Math Algorithms Numerical Issues Theory of Comp.	Discrete Math Design/Analysis Algo.	Discrete Math Algorithms or Theory
CS Electives (credits)	15	6	12	12	9
Lab Sciences (credits)	12		12	8	12

Table 7. Comparison of course requirements for the CS major at Queens, Brooklyn, City, Polytechnic University, and SUNY/Stony Brook. All programs for the BS degree degree.

Projections for Increases or Decreases in Enrollment

Consistent with the national trend, we had a 40% drop in enrollment following the burst of the internet bubble in 2001. However improvement has been evident in the past year, and it seems clear that we have passed the low point and are on the mend. Although we do not expect a repeat of the recent history, with hundreds of students rushing into our major in the coming years, we are confident that our enrollment will continue to move upwards since our discipline is at the heart as well as the forefront of the information revolution that is still in its infancy.

Plans for New or Revised Programs

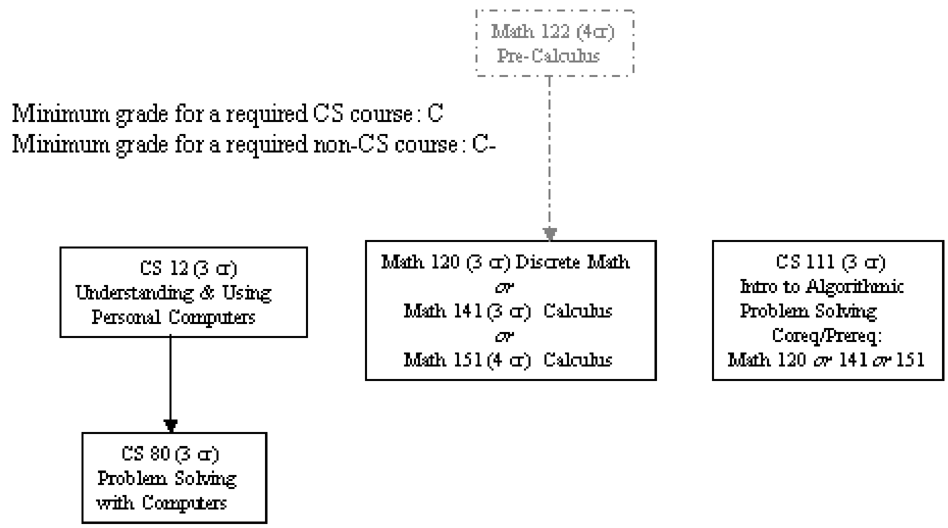
Our curriculum committee, with the involvement of the entire faculty, constantly monitors the latest developments in the field and devises improvements to our programs. A major revision ("Curriculum 2005") was introduced recently and is now in full implementation. We plan to review this curriculum and make necessary changes as we observe its effectiveness in the next year or two.

Curriculum: Programs for Non-Majors

We serve non-majors through three primary avenues. The first is our course CS12, Understanding Personal Computers, which we offer to the entire college community. Each semester hundreds of students take CS12 to satisfy the Scientific Methodology and Quantitative Reasoning area requirement of LASAR. The course can also be used to satisfy the computer requirement of such majors as accounting. A variation of the course, with business applications, is offered to students in the economics department as CS18.

Our second path for non-majors is our newly implemented minor in Computer Information Technology (CIT). This offering is designed to give majors in other fields practical experience with currently important information technologies and a sound understanding of the principles behind those technologies. As Figure 9 shows, the CIT Minor consists of CS12 and five CS courses (outside the major) that are more or less skill-oriented, plus CS111 and one of the math courses required by the major.

Computer Science Departmental Requirements — CIT Minor
and **Prerequisite Structure** (Curriculum 2005)



Four additional CS courses numbered 81-199 chosen from:

Database Application Programming
HTML and WWW Programming
Models of Computation
Multimedia Fundamentals and Applications
Science, Computing Tools, and Instrumentation
Topics in Computing

Figure 9. Prerequisite Structure for the CS Computer Information Technology Minor.

Finally, for those who want more rigorous CS training without majoring in CS, we have a CS minor consisting of nine courses. These courses constitute a subset of the CS and math courses required by the BA major, starting with CS111. Thus a student minoring in CS can easily switch his/her major to CS or add a second major in CS. Figure 10 shows the prerequisite structure of the CS Minor.

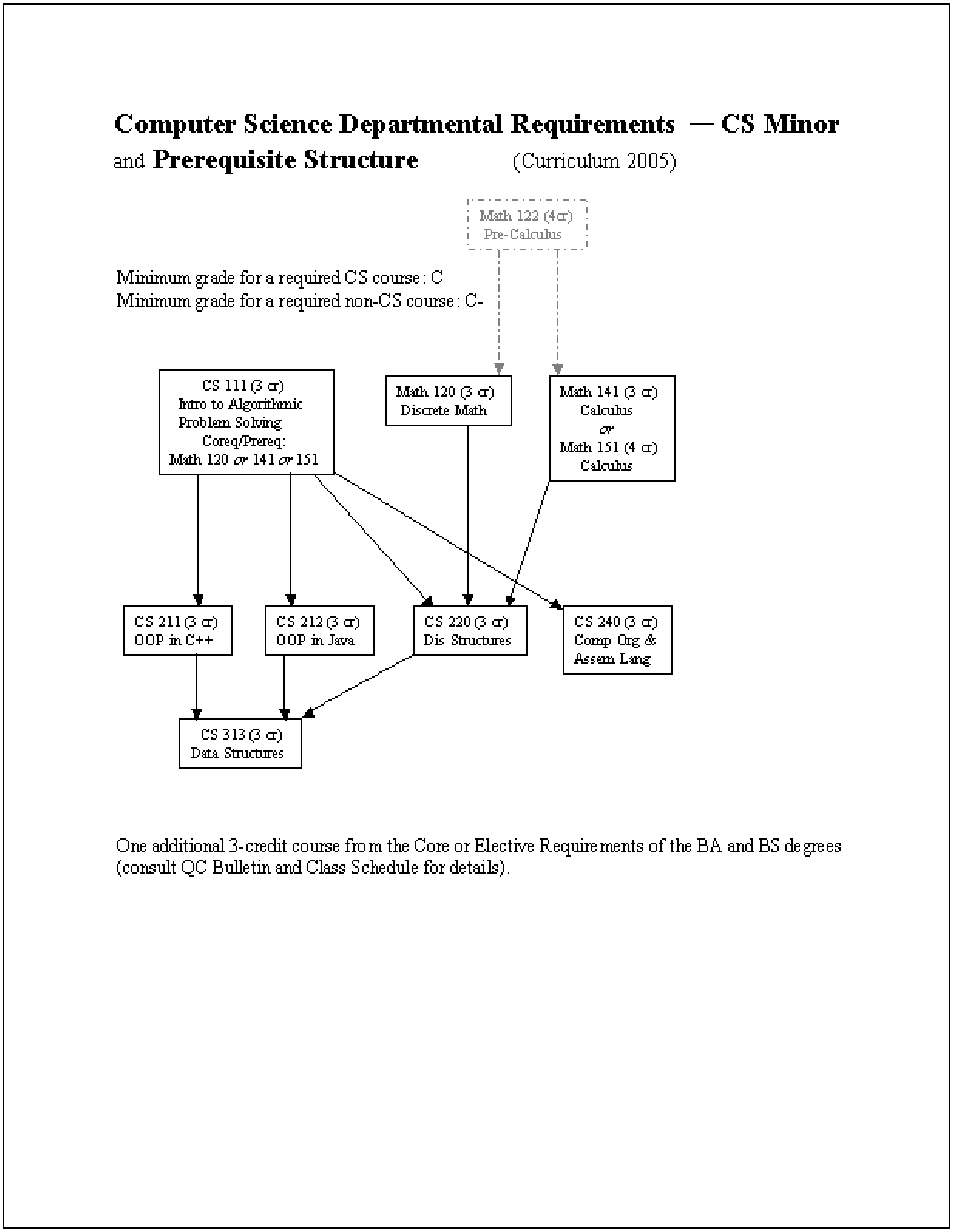


Figure 10. Prerequisite Structure for the CS Minor.

Master's Degree Program

Our Master's program consists of four *core* courses, a choice of three *semicore* courses, two *elective* courses, and a *capstone* course. The structure is given in more detail in Figure 11. Students with an undergraduate degree in some discipline other than computer science may enter the program, but only after they have completed six undergraduate courses that are essential components of our BA program. (These courses are shown in gray boxes at the top of figure Figure 11.) This allows us to accommodate students who want to make a career change or who are from foreign countries where programs in computer science comparable to those in this country were not readily available to them.

Several of our faculty, both established members of the department and our new hires, offer graduate-level special topics courses on subjects of current interest, such as bioinformatics, security, and wireless communication. We look forward to further strengthening this important part of our program in the future.

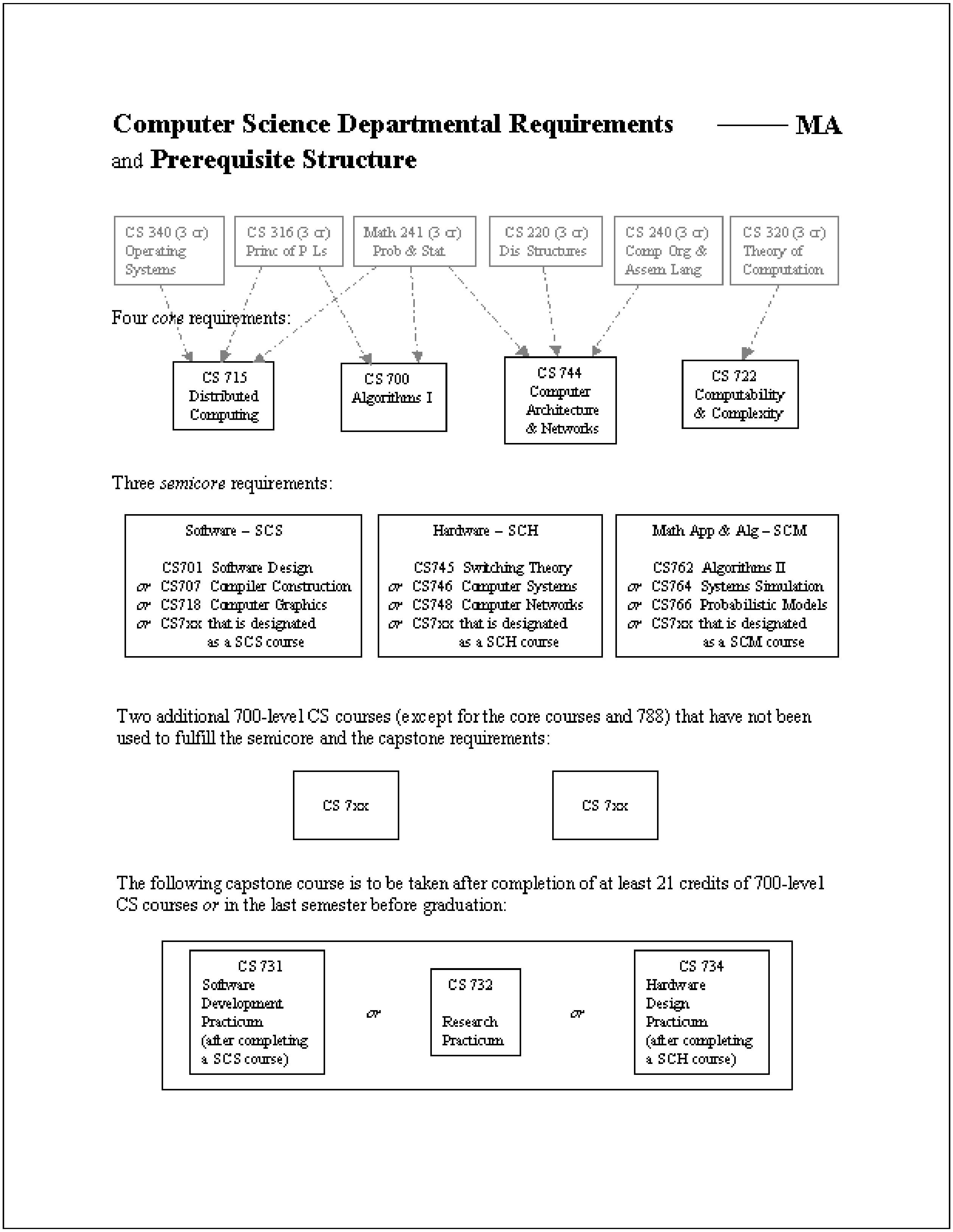


Figure 11. Prerequisite Structure for the MA in Computer Science.

Assessment of Effectiveness of Curriculum

Methods and Instruments Used

From the time of the department's inception, our curriculum has been based on national norms, starting with the curriculum known as "ACM '68" and continuing through to today's curriculum, which corresponds well to the current recommendations of the ACM and IEEE. However, we have always tailored our courses to meet what we perceive to be the particular needs of Queens College students and to reflect the particular strengths of our teaching faculty. To the extent that our curriculum implements national norms, we are confident that it accurately addresses the needs of computer science students in general.

We regularly review student course evaluations administered by the College's Teaching Excellence and Evaluation Committee to look for problems that may be occurring in our course offerings.

Student/Alumni Survey: In June of 2005 we conducted an online survey of our undergraduate students, graduate students, and alumni to gather more detailed information about the perceived effectiveness of our undergraduate and master's programs. The complete survey, with a numerical summary of the short answer questions, is given in Appendix C of this document. It is also available as a separate document (PDF): Survey Summary.

Measured Effectiveness in Achieving Desired Objectives

The survey was divided into separate parts targeted to current undergraduate students, current graduate students, undergraduate alumni, and graduate alumni. We had a total of 290 respondents to the survey, although 35 of these were discarded because they did not fall into one of our four target categories. Some individuals responded more than once because they fit into more than one category. In particular, many of the current graduate students were also undergraduate alumni.

The answer to Question 1 of the survey indicates that we received replies from 158 undergraduates, 41 graduate students, 39 undergraduate alumni, and 17 graduate alumni. While this is a gratifyingly large number of responses, especially from our current undergraduates, we should point out that there are certain sampling biases at work that affect the interpretation of the results. The first bias is that the survey was widely publicized among our current undergraduate and graduate students, but we did not have a way to contact alumni other than to put a note about the survey on the Department's home page where it might have been seen by alumni who happened to be visiting our web site. Thus, our alumni data probably draws most heavily on those graduates of our programs who are currently enrolled in the MA program and does not represent our alumni body at large. Furthermore, we discovered after the survey was closed that we made a mistake in constructing it. Based on the answer to the first question, participants were directed to one of four sets of questions based on their relationship to the department, or to a "thanks but no thanks" page if they were not current or previous students. Since we received no answers to the questions targeting graduate alumni, we infer that the 17 of them who answered the first question were misdirected to the "thanks but no thanks" page.

The first item of interest is the fact that 40% of the undergraduates who responded to the survey are transfer students from other colleges. Of these, over half transferred from a community college within CUNY. Our two biggest feeder colleges are Queensboro Community College and Laguardia College. We have anecdotal evidence that the preparation of these students varies greatly depending on which school they transferred from, and we recognize the need to provide meaningful articulation with the programs offered by the various CUNY community colleges.

Nine of the undergraduate respondents transferred from outside the US. Five transferred from other senior colleges within CUNY. Thirteen transferred from other four-year colleges in the New York metropolitan region (SUNY/Stony Brook, Columbia, NYU, St. John's, Adelphi, etc.)

Over half of the master's students who responded completed their bachelor's degree at Queens. Another 15% came from elsewhere in CUNY.

The full set of questions and summary data for short-answer questions are given in Appendix C. We summarize the data as follows:

- About one-third of our undergraduate students work full-time in addition to pursuing their studies. Another third work half-time, and the remainder do not work at all.
- Most undergraduates think it is reasonable to spend 2-6 hours a week outside of class on preparation for a 3-4 credit course, and that is also how much time they report actually spending. This result is consistent with the report of [Young 2002], indicating that college students no longer consider two hours of preparation for each classroom contact hour to be the norm.
- We asked the undergraduates to rate the program on sixteen attributes using a 5-point scale, with 5 being the best. The three highest ratings (over 3.6) were for the library and computing resources provided by the College, and for the relevance of the courses offered by the department.
- The lowest ratings (below 3.3) were for: preparation for work after graduation (3.01), reasonableness of time demands (3.16), proportion of courses taught by adjuncts (3.21), quality of advisement(3.26), responsiveness of the department to student needs (3.28), and availability of instructors outside of class (3.29). Since a score of 3.0 is equivalent to "Neutral," we feel that the results reflect very well on the department. Even our greatest perceived weakness (preparation for work after graduation) received a negative rating from only 26% of the respondents.
- We asked about the time demands of individual courses, but in only two cases did more than 10% of the respondents reply that the time demands for a course were "Too heavy!".
- More than 80% of our graduate students had undergraduate majors that were in computer science or a related discipline.
- The graduate students' ratings were generally lower than the undergraduates'. Our highest approval ratings were for fair grading, good preparation for further graduate study, course relevance, computing and library resources, and faculty expertise, but these ratings ranged only from 3.3 up to 3.5. Weaknesses (average ratings below 3.0) were choice of courses, preparation for work after graduation, availability of internships, and responsiveness to student needs.
- With only 36 graduates of our undergraduate program responding to the survey, the data become tenuous. Of these respondents, over 60% reported that they had pursued no formal education beyond the bachelor's degree. Over a quarter of the graduates who had done graduate work had partially completed a master's degree in a computer related field; we infer that these are our own master's students.
- We asked the students to describe their employment history since graduation, and received the following responses:
    - 50% had been regularly employed in jobs related to computing.
    - 22% had been regularly employed in jobs not related to computing.
    - 19% reported having multiple jobs, some of which were computing related, and some of which were not.
    - 8% had not been employed since graduation.
- Over half of the graduates of our undergraduate program reported that their time spent at Queens College was "rewarding and enjoyable." Most felt that the computer science major provided good preparation for work, but over a third disagreed with this assessment of the program.

We also asked respondents to give written answers to some questions. We asked our current undergraduates to tell us the two most important things they thought the Computer Science Department should do in order to improve (survey question 12). We received 78 answers to this question. By far the most common comments (24 of them) dealt with a perceived need for improved pedagogy, including the need for tutors, teaching assistants, and gripes/kudos for individual faculty members. We also received 15 responses dealing with the content level of the courses; these were about equally divided among those who thought the courses should be easier (5 responses) or harder (3), with the remainder asking for particular subject matter to be included or omitted. There were also several replies that expressed a desire for better scheduling of courses so that there would be fewer scheduling conflicts.

We also asked the alumni of our undergraduate programs what they felt the Department and the College should do to improve. Because of the way the survey was publicized, most respondents were also current students, so we also include responses by current MA students, who were also asked what the Department should do to improve. Far and away, the largest cluster of comments dealt with the perceived desire for more job-oriented, real-world skills, and more internships. Other comments generally reflected the same perceived shortcomings found in the short answer questions, mentioned above.

Use of Assessment to Improve Programs

The survey findings were quite consistent with the informal perceptions we have had about ourselves as a department from the time of our inception: We feel that we provide high-quality instruction in the core principles of computer science, but that many of our current students and recent graduates would prefer vocational training in information technology. It is our firm belief that the main benefit of our curriculum is that it provides students with the basis for life-long learning, and that to devote a significant amount of the curriculum to current "hot topics" leaves students ill-prepared for the long haul. (Even reviewing the particular topics students wanted us to teach about six months ago reveals an interest in "older" technologies, such as J2EE and Python, but no mention of current technologies such as Ajax and Ruby.)

For a long time, we tried to convince students that their interest in a BS degree was misguided and that the BA is the preferred degree from a liberal arts college such as Queens. Finally (over a decade ago), without substantially altering the degree requirements, we introduced the BS option. Students have flocked to it, and the department has been able to maintain its own perceived integrity of the curriculum at the same time. We take this case to be proof that the difference between students' perceived needs and the department's sense of mission do not have to be irreconcilable.

Currently, there is a large demand for courses that more properly fall under the information technology part of the computing umbrella, rather than computer science. We feel that we cannot incorporate this area into our major without losing focus on core principles of computer science. But we have been able to meet student interests by introducing our minor in Computer Information Technology (CIT). This minor is too new to evaluate at this time, but indications are that it will be a very popular option both for students majoring in other disciplines as well as some of our own majors. We expect enrollments in this minor to be even more volatile than those in our majors because of the vagaries of the perceived job market.

We are enthusiastic about the CIT minor for two reasons worth mentioning here: (1) The simple fact of its existence helps us explain to students (and colleagues) the difference between computer science and information technology. There has been a public relations problem leading to misconceptions about the department and its mission in the past, and the CIT minor helps us to articulate the nature of the department more clearly to others. (2) The CIT minor also provides a nice opportunity for our faculty to demonstrate their involvement with real-world problems. Research in computer science is often driven by issues arising from the practical applications of computing theory and principles to current technology. The CIT minor is a means for us to integrate some of our teaching with the issues that are driving computer science research today.

Students

Undergraduate

Diversity of Demographics

As reported above, about 40% of our undergraduate students are transfers from other colleges and of these, half are from community colleges within CUNY. The rest of the transfers come from other senior colleges within CUNY, other metropolitan area schools and, a not insignificant number transfer from colleges abroad. The other 60% come to us mostly after graduating from local high schools in the New York metropolitan area.

The ethnic and economic backgrounds of our majors generally reflect the diversity of the college as a whole. There are differences, however. Where the undergraduate gender ratio at the College is over 60% female, we estimate that the ratio is reversed in the Department, and that over 60% of our students are male. The undergraduate ethnic distribution for the College is 50% White, non-Hispanic, 22% Asian or Pacific Islander, 17% Hispanic, and 10% Black. We believe that reversing the College's percentages for the White and Asian categories would more closely approximate the Department's distribution.

Career Choices and Perceptions of Program

An important criterion in assessing the success of our program is the extent to which our graduates are able to secure appropriate professional employment. Five years ago the majority of them were able to find IT related positions quite easily. In fact, many students had multiple job offers to choose from. However the dot-com bust and the increasing trend toward outsourcing lower-level projects, has had a significant effect on the job prospects of graduating computer science students nationally. Employers are still hiring, but with fewer positions available and a large pool of candidates, it's pretty much a buyer's market. Given these circumstances, the fact that half of our graduates have found computer related jobs (see Assessment of Effectiveness of Curriculum) should be considered a significant accomplishment. The trend toward hiring high-quality talent means that half of our students were judged as such by the market.

It should be noted that the set of possible job categories for graduates of our program is quite large, and personality factors as well as skill plays an important role in the path eventually pursued. Assuming that a student is temperamentally suited for a particular type of job, different jobs require different skill levels, knowledge, and overall intellectual ability, and students seem to be quite good and determining where they stand with respect to these factors. We find that our most academically successful students tend to seek jobs as programmers in software companies (working in-house or as consultants) and, with increasing frequency, in the financial services sector. Second tier students are usually more interested in database and systems administration positions or in work as "business analysts", network specialists or in technical sales support. Those who can't find employment in the above two areas will gravitate toward "help desk" jobs, work in PC support or in lower-level networking and systems administration positions. While the above is clearly a generalization, it is in the main correct. Where do our students get the knowledge needed to pursue these positions and how did our curriculum contribute to this success?

Overall the undergraduates rated the relevance of our courses quite highly. The undergraduate computer science curriculum is designed primarily to give our students the tools to become first rate programmers and systems analysts. It emphasizes fundamental principles as well as provides for the teaching of a number of particular technologies. Our graduates find that much of what they need to know when they start their careers has been covered in one or more of their courses. If their job requires that they master something that they did not explicitly cover in their coursework, many report that the principles that they learned allow them to acquire new skills with confidence.

While the above speaks to the students' performance once they get a job, an especially important factor in the students' ability to land that all important first position is whether or not they are able to get an internship while still in the program. This is the case for both undergraduate and graduate students. Potential employers consider a successful internship as an indication that the student is serious, industrious and able to function in the business world. In light of this, the department has stepped up efforts both in cooperation with the job placement office and through our own contacts to increase the number and quality of internship opportunities available to our students.

Not all our students go directly into industry, however. In recent years we have also seen a number of our students go on to pursue graduate studies at some of the top-ranked Computer Science programs in the country, including Princeton, Carnegie Mellon, UC Berkeley, Columbia, and NYU. In addition, we regularly send students to the highly-regarded regional programs at the CUNY Graduate Center and SUNY Stony Brook.

Graduate

Our graduate enrollment has dropped by about two-thirds since 9/11. This decline is attributable to the difficulty many foreign students have in obtaining visas to study in the US as well as the general drop in computer science enrollments nationwide. Though the number has declined, we believe that the quality has actually improved. This is very likely attributable a change in the backgrounds of our current students. We previously admitted many students who wanted to change majors for the graduate degree and whose preparation in computer science was weak or non-existent. Those students were required to take a complement of core undergraduate courses in the Department in order to prepare them for graduate-level work. But now more than 80% of of our graduate students have had undergraduate majors that were in computer science or a related discipline, and many have been our own undergraduates at Queens College.

Despite the drop, our graduate enrollment has been growing for the past three semesters for which we have data (Fall 2003, Spring 2004, Fall 2004). The graduate students give various reasons for enrolling in the program. Many believe that additional training at the master’s level will help them get a good job in this tough employment climate. In fact, students with a master’s degree tend to find it easier to get a job, and when they do they got a slightly higher starting salary and more interesting work. Others enroll in our program because they find our wide selection of courses very appealing and useful. Indeed, over the last three years we have added courses in such important emerging areas as cryptography, Internet and web technologies, Bioinformatics, Mobile Computing, Next Generation Network Services, Voice over IP / WLAN, and Internet Security. Aside from their intrinsic interest, knowledge of these topics is sought after in industry today. Such a wide variety of cutting edge courses is quite rare and these offerings have been an important factor in helping our graduate program grow.

We have also made a change in our admissions procedure in the past few years that we think contributes to our improved enrollments. Where we previously processed graduate applications just once a term, we now handle them on a “rolling admissions” basis. This means that qualified students are admitted as soon as we have their application in hand, reducing the problems students have arranging visas, work, and travel plans. The result has been positive: in 2001, just 28% of the students we admitted actually matriculated into the program. In 2005, the proportion was up to 63%. Despite the drop in absolute number of students, we remain the largest master’s program in the division and are cautiously optimistic that our graduate enrollments will continue growing at a reasonable rate.

Self-Analysis and Plans

Strengths

The first and most important resource of any academic department, the core of its strength, is its faculty. The Department of Computer Science is proud of the professional standing and the distinguished research record of its members. Members of the department have been active mainly in the following research areas: information retrieval, cryptography and computer security, visual information processing, fundamental mathematical theory of computation, computer architecture, web development and finally a new area for the department, bioinformatics. In computer science research, many specialized areas tend to gain and lose centrality as the subject evolves. The faculty of our department have very successfully adapted to this situation, continually moving on to new and developing areas of research.

This Department has always emphasized undergraduate instruction, regarding this as one of the core missions of the College. This role is especially important in computer science, because of the special character of undergraduate education in a rapidly evolving field. In most of the sciences a post-graduate degree is a necessary minimum prerequisite to begin a professional career. This is not true in computer science. Many of our graduates begin a professional career with a BA or BS, and are able to continue their professional education while working in technological industries. Because computing technology develops so rapidly, essentially all computing professionals are obliged, if they wish to remain in the field, to pursue a lifelong process of continuing education. The undergraduate program our Department offers is designed to give them a solid foundation for this process. Members of our faculty have also devoted great efforts to curriculum development and mentoring undergraduates.

The Department has also developed a program in computer literacy that it offers to all members of the college community. Our introductory course, CS 12, has enrolled hundreds of undergraduates every year. We are proud of the record of success this course has achieved. It has become a core course in the education of many undergraduates, from diverse subject areas.

This semester the Department has completed the introduction of a new minor in Computer Information Technology. This June we will be celebrating the graduation of the first students concentrating in this area. This program emphasizes applications of computing including web programming, database, and multimedia design.

The master’s program is a central program of the Department. In the recent past most of the students in this program had a bachelor’s degree from other areas, and were retraining in computer science. The Department developed an extensive sequence of courses designed to rapidly bring these students up to the level required for a traditional master’s program in computer science. In the last three years, the department has greatly enhanced its graduate course offerings, and in consequence been able to attract many more master’s students with an undergraduate CS degree. This has strengthened the master’s program considerably.

The Department also benefits from its excellent and dedicated technical staff. Peter Chen, our network manager, and Koya Matsuo, our system administrator, have developed and maintained our computing and network resources, despite having to work within the constraints of a very a narrow budget. The Department is also grateful to the Queens College administration, which has provided extensive computing facilities for the use of its undergraduates.

Weaknesses

The Department is acutely aware of certain areas of weakness, and is dedicated to solving the problems they represent. Many of these weaknesses arise ultimately from the unavoidable constraints imposed by the College budget.

It has proved impossible for the department to hire faculty in certain very competitive areas. In the more competitive areas of computer science, such as computational genomics and computer security, starting salaries are at least 50% above the level the College is able to offer, and typically come with start up monies far beyond the level we can approach. Recently, the Department was able to hire in the area of bioinformatics by working with the Biology Department and hiring a very highly qualified person from the more biological side of bioinformatics. A major goal of the department is more hires in this vital and growing area, in order to develop a functioning research group.

Similarly in computer security and cryptography, the salary scale of the College has restricted the types of candidates the department can recruit. Here too, the Department was able to hire very successfully, recruiting an exceptionally talented individual with a less applied background, who had trained in analytic number theory and had previously worked at the National Security Agency.

This general strategy has been a foundation of the department's hiring policy. Since the more theoretical areas of computer science generally command lower salaries, the department has tried to hire outstanding individuals whose research areas are on the more theoretical side, but who are also capable of teaching the very applied courses which the department's undergraduate curriculum requires.

However, in emerging research areas, it is vital to be able to hire at least one senior faculty member who can function as a research group leader and mentor to junior faculty. Such an individual can take the lead in organizing external grant applications and collaboration with research groups at other institutions. They can also organize and direct curriculum in development in new areas. Unfortunately, the department has been unable to hire such vital individuals. Such hires could radically transform and improve our department.

The department also acknowledges that so far it has not been able to develop the relationships it envisions for itself with neighboring institutions, such as Rockefeller University and Cold Spring Harbor Laboratory. The Department is pleased by its developing relationship with Computer Associates, but acknowledges that it needs to greatly enhance its relationship both with this company and with other regional technology companies. Of course, developing these relationships will require some investment of resources, both in funds and in released time for the faculty who undertake these duties.

The Department is grateful to the College for the released time which has been granted to new hires. However, the Department has been unable to offer its other faculty adequate released time for research. This gap is especially damaging in computer science, where faculty must continually update and revise courses in order to keep up with rapidly evolving technology. In a field such as mathematics, many undergraduate courses present material which has been fundamentally unchanged for over a century. In computer science, especially in software, a course which is not extensively revised every three years is badly out of date. It is unthinkable that we would run a software course the way it was run in 1990.

Proposed Changes

Department Organization

The Department envisages two major changes in its structure and organization which it would like to implement.

Firstly, the department needs to hire at least two senior faculty who can function as research group leaders in each of bioinformatics and computer security. Such individuals can organize grant applications and mentor both junior faculty and advanced students. They can organize curriculum development and lead in setting up collaboration with neighboring institutions. They also relieve junior faculty of the research isolation which can so severely inhibit their productivity.

Secondly the department intends to expand and solidify its new technology minor. This program is an important resource for the College, and offers a career path for many students in other departments who are not now served by the department's courses for its majors. In addition it will help to even out fluctuations in the department's student enrollments, which tend to reflect the job market in the computer industry.

Programs and Curriculum

We would divide our curriculum into macro and micro levels. At the macro level, our curriculum is quite stable in that the core principles on which computer science is based have not altered greatly since the earliest days of programmable digital systems in the mid-twentieth century. As with any science, the core of the discipline generally changes slowly, and we do not often need to make drastic changes to the set of courses we offer in order to track these changes.

On the other hand, the micro level of the curriculum is has always been under active revision from the day the department was founded over thirty years ago. At the micro level we deal with issues like like what programming language to use for projects, what computing environment to work with, and what are the current best practices in software design. These changes typically don't show up in the College Bulletin, but they do consume a considerable portion of our curriculum development activities.

Where we do see macro-curricular developments taking place over the next five to seven years is in the area of cross-disciplinary interactions with other departments on campus. We are already working with the Biology and Economics departments to provide courses that will give their students the technological understanding they need, and we anticipate that this sort of activity will play an increasingly important part in our role on campus.

Our curriculum provides an essential part of a liberal arts education in the 21st century, and one of our goals is to project an accurate image of that role, one that avoids hype and job-market frenzy in favor of recognition of the core role a working knowledge of computing plays in an educated person's life.

Resources

The Department has outgrown its office and research space allotments and must deal with these issues for both short-term and long-term time frames.

We are currently providing new faculty with windowless offices and research space that must be shared with instructional labs or departmental servers. Rather than deal with these problems in an *ad hoc* case by case basis, we would like to plan now for the addition of six additional faculty offices that are of the same quality as most of our present offices with respect to space, light, and access to departmental networking facilities. The most natural location for additional office space would be on the third floor of the "A" wing of the Science Building.

Two main research foci, bioinformatics and information security, need research spaces that address their needs in terms of workstations and meeting facilities. Our work in Web services needs its own research space in addition to the educational lab being established in A-103 from Technology Fee funds, and we have no facility for our work on embedded systems and small-area networks. The most natural location for these laboratories would be on the first floor of the "A" wing of the Science Building. Additional research space is critical to our ability to obtain outside funding for our research efforts. However, funding agencies have not normally supported the establishment of an institution's research infrastructure, so it is imperative for the College to obtain the needed funding for converting classrooms to research laboratories from alternate channels.

Future of Department for Next Five to Seven Years

Hiring

The central element of the Department's mid-range planning is its hiring strategy. Over this time period, the Department approximates it will have two retirements, with corresponding replacement faculty lines. Additionally, we expect to obtain some new lines in the near future.

The Department's faculty pursue fundamental research in many areas of computer science. But, in addition to this core research, members of our faculty have been engaged in collaborative research with faculty from many other departments. We expect a great increase in opportunities for such collaborative research in the years ahead, as computational methods invade new fields. Biology is witnessing the rise of sub-disciplines, such as computational genomics, which apply sophisticated methods of mathematical and computational analysis to vast bodies of data that are now available. Computational revolutions are also taking place in other sciences, because of the enormous growth in recent years in the processing power and memory size of computers that are readily available to researchers. The Department welcomes its role as a collaborator with other departments in the College.

We have concluded that the Department will make the most effective use of its resources if it attracts and hires researchers in cluster groups, especially where it can take advantage of interdisciplinary collaboration with allied departments. To this end, the Department has identified the following two areas where it would like to concentrate its resources. These areas have been chosen not only because they are of much current interest in our discipline, but also because of their intrinsic scientific importance, and the potential contribution they would make to the College as a whole.

Bioinformatics: This field combines methods of biology and computer science, especially in areas where new sequencing technology has produced an enormous volume of genetic data that provides unprecedented opportunities. For example, it is now possible to determine a huge portion of the genomic variation in thousands of patients in a disease study. But the torrent of available data also presents great computational challenges. The Department would like to collaborate with the Biology and the Chemistry and Biochemistry Departments to build a significant research group in this area. We believe it is crucial to hire at least one senior scientist--someone who will be a group leader, organize and lead grant applications, attract and mentor younger co- workers, and effectively foster interaction with others in the bioinformatics community (which we believe is important, as a sense of isolation can greatly inhibit the productivity of young scientists). In addition, we envisage hiring junior faculty in this area.

Information Security: This is another area of great scientific interest that has very important applications. The Department already has two members in this area, one in cryptography and one who works in information assurance. Our development plan is to hire one senior scientist in this area and additional junior faculty members.

Hiring in both of these core areas would bring many advantages to the wider College community. Both fields are inter-disciplinary, and would energize collaboration with allied departments. Both fields offer students many opportunities for employment after graduation and for graduate study, and both are very rich in research opportunities for undergraduates.

In addition to hiring in these two core development areas, the Department hopes to hire other faculty members. The specializations we are most interested in now are architecture, programming languages, and human-computer interaction. However, past experience has taught us that over a seven-year period new areas of importance are likely to emerge.

Curriculum Development

The Computer Science curriculum is in a constant state of redevelopment and evolution, partly because the lifespan of a technology era is rarely as long as ten years. Nevertheless, the basic content and structure of the major, which is largely dictated by norms developed by professional societies in the field and other academic institutions, may not change too much over the next five to seven years. One significant change we plan to implement over this period is to offer more undergraduate courses in the above- mentioned areas of bioinformatics and information security. We hope to develop a strong sequence of graduate courses in each area as well.

A master's program in bioinformatics is currently in the planning stages with the Biology Department. We have both new hires and established faculty who are actively involved in various aspects of this exciting discipline, and we have established good rapport with several members of the Biology faculty who have complementary interests. We feel that a master's program in this field will draw well-qualified students in the region.

The Department also hopes to develop a collaborative undergraduate program with the Economics Department in computational finance, linked to the BBA program.

In addition, the Department plans to strengthen and expand both its Computer Information Technology (CIT) minor and its service-course offerings for non-majors. The recently-introduced technology minor provides instruction in the design and deployment of internet-based applications. While courses in this minor could be taught from an Information Technology perspective and stress the use of commercial tools, our courses stress design principles and realistic models based on algorithmic problem-solving. We believe that this minor will be seen by students as relevant and immediately stimulating, and will also motivate some of them to dig deeper and become computer science majors.

In an era where technological skills and a sound understanding of technological issues and trends are increasingly expected of college graduates, we feel our curriculum must address the needs of the Queens College student population--not just the needs of Computer Science majors. This is one reason why it is important for us to expand our CIT minor and other course-offerings for non- majors.

A second reason is that this is a good way for us to increase and stabilize our enrollment. Historically (if one can speak historically of the youngest discipline in our division!), Computer Science has been a "big major," with around sixty credits of required courses, and a "big department," with the fourth largest number of majors at the College in 2003. But, as "the technology department" at the College, Computer Science attracts many students who pursue their undergraduate education with a quite limited, career-oriented objective: to prepare themselves for their first job. While this relates to only one part of the Department's mission, it has led to a large fluctuation in our enrollment over the past few years as the regional technology-job market has fluctuated. From 2001 to 2004 our FTEs dropped by close to 40% -- reflecting the "dot-com bubble burst" of 2001-02 and subsequent concerns about off-shoring of programming jobs -- though more recent enrollment figures indicate that we are now past the low point and the number of students majoring or minoring in our subject is moving up again. We expect the planned expansion of our technology minor and service-course offerings for non-majors to significantly increase our student numbers, both directly and by attracting more well-qualified students to pursue our major, and so provide a more stable basis for the Department's future growth.

Projecting Our Identity

The Department needs to make its strengths and role on campus better understood by two audiences: prospective students, and our peers in other departments.

It is important for us to make clear to prospective students how and why programs in computer science, and our majors/minors in particular, are so important in today's liberal arts education. Students need to understand the five computing sub-disciplines identified by the IEEE/CS report cited in the Introduction, and how our department relates to those disciplines.

We believe that our peers in other departments are generally very much aware of the central role that technology plays in today's society and how it affects their own disciplines. Still, we feel it is important to make it clear that, in addition to using technology effectively, the College needs to integrate an understanding of technology into the core of its curriculum, and that the Computer Science Department is the entity that is uniquely qualified to serve as the focal point of this integration process.

Recruitment and Retention of Faculty and Staff

Faculty

The Department has been fortunate to attract young and talented faculty members for tenure-track Assistant Professorships. Since Fall 2003 the following faculty members have joined the Department: Jinlin Chen (web design), Kent Boklan (cryptography), Jun Zheng (computer architecture) and Boojala Reddy (bioinformatics). We have been able to compete in the computer science job market, despite lower starting salaries, by offering start-up packages that boost our offers: equipment budgets up to $80,000, travel money for the first two years, and summer money for one or two months after the first academic year of employment. In addition, the College offers new faculty 12 hours release time over the first three years. We believe that we must continue to offer salaries at the top of the CUNY Assistant Professor scale to attract the best candidates, especially in highly competitive areas such as bioinformatics and information security.

Two faculty members have left the Department over the last three years: Carol Friedman (biomedical informatics) joined the Columbia Biomedical Informatics Department in Summer 2004 as a tenured full professor. Mingzhou Song (bioinformatics) joined the Bioinformatics program at New Mexico State University in Fall 2005 as an Assistant Professor.

The College was extremely supportive in our attempts to retain Carol Friedman, offering attractive salary and release time incentives. Although Columbia ultimately made it too difficult for Carol to return to us, we see that the College has demonstrated its willingness and ability to help us attract and retain excellent faculty insofar as it is possible.

Although most of our faculty have remained in the Department and obtained tenure, it is clear that faculty in some sought-after areas such as bioinformatics will have other opportunities: larger salaries, lower teaching loads, and more resources.

Technical Staff

As mentioned elsewhere, the Department must of necessity rely on its own technical staff to manage and maintain our networking and computing infrastructure. The College's Office of Converging Technologies is not positioned to provide the level of technical expertise and responsiveness we need to deal with either our day to day or our long-term requirements.

We currently have two full-time technical staff (Peter Chen and Koya Matsuo), and one part-time college assistant (Xiuyi Huang). Chen manages the Department's network infrastructure, Unix workstations, and handles administrative duties such as budget preparation and purchasing as well. While we are extremely fortunate to have such a talented and hardworking individual providing these services, the Department is so dependent on his services that we feel vulnerable because so much of our operations are so dependent on this single individual.

Matsuo can provide backup for Chen in some day-to-day network management tasks, but his real talent is in managing the department's numerous Linux and Windows systems. Huang provides support for application software and web development for the department. She also provides some support for hardware management. Although she is a skilled and dedicated resource for the Department, we are constrained by the fact that she is not a full-time employee.

An area where we feel we have been deficient is in providing professional development resources for our technical staff. They need to be funded for ongoing training so that they can continue to provide the department with the flexible state of the art infrastructure that we have come to depend on. Because there is no realistic alternative to being self-sufficient with regard to planning, installing, and maintaining our technical infrastructure, it is important that we develop as much cross-training among the members of our staff as possible in order to minimize our exposure to overdependence on single individuals.

Questions for and Advice Sought from External Reviewers

- We feel that the main problem facing the department is its high teaching load. Do you agree?
- Given that tenure and promotions are based on research productivity, how can faculty balance their teaching duties and their research needs?
- Is there a precedent we can cite to convince the administration to take into account the fact that teaching in computer science often requires more preparation than teaching in most other disciplines, owing to the continual and rapid evolution of our field?
- We believe that the Department needs to establish research leadership in cluster areas in order to improve external funding. Do you agree?
- Do you think the areas we have chosen for growth, bioinformatics and information security, are appropriate given the current state of the discipline and our current faculty's credentials in these areas?
- What actions should the department take to attract leaders in these areas?
- What other steps should be taken to increase our research productivity and external funding?
- Are there changes we should make to improve our curriculum?
- How can we take advantage of the fact that we are located in New York City and the unique resources it provides?
- Do you see other weaknesses we are overlooking, but should attend to?
- Do you see ways in which we could better exploit our strengths?

Web Links

Association for Computing Machinery (ACM)
www.acm.org

Computer Research Association (CRA)
www.cra.org

Institute for Electronics and Electrical Engineers
www.ieee.org

Intel Corporation
www.intel.com

National Education Association (NEA)
www.nea.org

Queens College
qcpages.qc.cuny.edu

Safari Bookshelf
safaribooksonline.com

References

[ABET 2005]	ABET Corporation	Criteria For Accrediting Computing Programs Effective for Evaluations During the 2006-2007 Accreditation Cycle. *Baltimore: ABET Corporation, October 2005.* *Online link*
[ACM 2005]	Joint Task Force for Computing Curricula 2005, R. Shackelford (Chair)	Computing Curricula 2005: The Overview Report. *Online Link*.
[QC 2005]	Office of Institutional Research	Queens College Fact Book 2004-2005. *Queens College Office of Institutional Research, 2005.*
[Vegso 2005]	Vegso, J.	Interest in CS as a Major Drops Among Incoming Freshmen. *Computing Research News, 17(3), May 2005. Online Link*.
[Young 2002]	Young J. R.	Homework? What Homework? *Chronicle of Higher Education, December 6, 2002. Online link*
[Zweben 2005]	Zweben, S.	2003-2004 Taulbee Survey: Record Ph.D. Production on the Horizon; Undergraduate Enrollments Continue in Decline. *Computing Research News, 17(3), May 2005. Online Link*.

Appendices

- Appendix A: Course Descriptions
- Appendix B: Faculty Vitae
- Appendix C: Student/Alumni Survey

Appendix A: Course Descriptions

12. Understanding and Using Personal Computers. 2 lec., 2 lab. hr.; 3 cr. Prereq.: Two and one-half years of high school mathematics, including intermediate algebra, or Mathematics 6 or 8. Hands-on introduction to computers, computation, and the basics of computer hardware and software. Students will have experience during the instructed microcomputer lab with a number of software environments including an operating system, a word processor, a spreadsheet and a database package. The course will focus on problem solving and programming with the context of these packages. In addition, students will acquire the skills needed to learn other software packages on their own. Not open for credit to students who have taken Computer Science 18. (SQ)

18. Computers with Business Applications. 2 lec., 2 lab. hr.; 3 cr. Prereq.: Admission to the Business and Liberal Arts minor or the Business Administration major. Fundamentals of using the operating system and application software. Business-oriented uses of software applications including: word processing, spreadsheets, presentations, and database management. Emphasis on realistic situations and problem-solving strategies used in business. An important part of the course is a research project/presentation of topics involving current issues arising from the use of computer technology in a business environment. Some sections will be limited to those admitted to the major in business administration, and others will be limited to those admitted to the minor in Business and Liberal Arts (BALA). (SQ)

80. Problem Solving with Computers. 2 lec., 2 lab hr.; 3 cr. Prereq.: Computer Science 12. An introduction to computer science through problem solving, focusing on the methodology of problem solving rather than specific hardware or software tools. Students will learn how to select and use specific software tools advantageously. Lab exercises will exemplify the problem solving methodology. (SQ)

81. HTML and WWW Programming. 3 hr.; 3 cr. Prereq.: Computer Science 80. Introduction to computer networks from a user's perspective and the World Wide Web. The course will provide hands-on experience with electronic mail, file transfer, Telnet, and Web browsers, including the creation of Web pages using HTML, Javascript, and CGI scripts; image preparation and editing; scanning and OCR.

82. Multimedia Fundamentals and Applications. 3 hr.; 3 cr. Prereq.: Computer Science 80. A comprehensive introduction to the fundamental concepts, techniques, and tools that underlie the use of multimedia in scientific and business applications. Major topics include the principles of image, sound, and video synthesis; software and industry standards; and typical applications.

84. Models of Computation. 3 hr.; 3 cr. Prereq.: Math 122. This course is intended to develop the ability to solve problems using differing models of computation. It will develop reasoning ability by creating a computing environment with very few rules which will then be used to develop algorithms within the scope of the model of computation. These environments will be models of actual computing environments. The nature of what an algorithm is will be developed.

85. Database Application Programming. 3 hr.; 3 cr. Prereq.: Computer Science 80. A continuation of Computer Science 80. Students will learn to program databases using SQL. Microsoft Access integrated with Visual Basic. In addition, object-oriented database programming such as Oracle and Jasmine will be covered.

86. Science, Computing Tools, and Instrumentation. 4 hr.; 3 cr. Prereq.: Math 122. Science and society; principles for scientific exploration; scientific visualization and mathematical analysis: concepts and techniques; computing tools for visualization and computational analysis; Internet tools for science exploration; concept of integrated computing environment for scientific study and collaboration; PC-instrumentation. Applications to social science, biochemistry, psychology, physical, chemical, and earth science. (SQ)

90.1, 90.2, 90.3. Topics in Computing. 1 hr.; 1 cr., 2 hr.; 2 cr., 3 hr.; 3 cr. Topics in computer programming and applications at a level appropriate for students who are not majoring in computer science.

Topics and prerequisites will be announced at registration time. The course may be repeated for credit providing the topic is different, and may not be applied toward the major in computer science.

111. Introduction to Algorithmic Problem Solving. 2 lec., 2 lab. hr.; 3 cr. Prereq. or coreq.: Math 120 or 151 or equivalent. Introduction to the principles and practice of programming. Topics include primitive data types; concepts of object, class, and method; control structures; arrays; procedures and functions; parameter passing; scope and lifetime of variables; input and output; documentation.

112. Introduction to Algorithmic Problem Solving in Java. 2 lec., 2 lab. hr.; 3 cr. Prereq. or coreq.: Math 120 or 151 or equivalent, and open only to students in the TIME-2000 program (consult the Department of Secondary Education for details). Introduction to the principles and practice of programming. Topics include primitive data types; concepts of object, class, and method; control structures; arrays; procedures and functions; parameter passing; scope and lifetime of variables; input and output; documentation.

211. Object-Oriented Programming in C++. 2 lec., 2 lab. hr.; 3 cr. Prereq.: Computer Science 111. Object-oriented algorithmic problem solving in C++, with attention to general as well as language-specific issues including pointer and pointer arithmetic; linked lists; memory management; recursion; operator overloading; inheritance and polymorphism; stream and file I/O; exception handling; templates and STL; applications of simple data structures; testing and debugging techniques.

212. Object-Oriented Programming in Java. 2 lec., 2 lab. hr.; 3 cr. Prereq.: Computer Science 111. Object-oriented algorithmic problem solving in Java, with attention to general as well as language-specific issues including applications, event- driven programming; elements of graphical user interfaces (GUIs); linked lists; recursion; inheritance and polymorphism; file I/O; exception handling; packages; applications of simple data structures; applets; concept of multi-threading; testing and debugging.

220. Discrete Structures. 3 lec. hr.; 3 cr. Prereq.: Mathematics 120 and 151 or 141; Computer Science 111. Algorithms, recursion, recurrences, asymptotes, relations, graphs and trees, applications. (SQ)

240. Computer Organization and Assembly Language. 3 lec., 1 lab. hr.; 3 cr. Prereq.: Computer Science 111. Principles of computer design and implementation. Instruction set architecture and register transfer level execution; storage formats; binary data encoding; bus structures; assembly language programming. (SQ)

280. Self-Study Programming. 3 hr.; 1 cr. Prereq.: Computer Science 313. Self-study and mastery of a programming language or package through reading and practice. Students should consult the department at the beginning of the semester for reading materials and assignments. May be repeated for a maximum of five credits provided the topic is different.

310. WWW Programming. 1 hr.; 1 cr. Prereq.: Permission of the instructor. Students will learn to do server-side programming for Web pages through hands-on assignments. Topics include the Common Gateway Interface (CGI), UNIX scripts in PERL, Javascript, image manipulation, and text scanning. May not be used as an elective for the computer science major.

313. Data Structures. 3 hr.; 3 cr. Prereq.: Computer Science 211, 212, and 220. Fundamental data structures and their implementations: stacks, queues, trees (binary and AVL), heaps, graphs, hash tables. Searching and sorting algorithms. Runtime analysis. Examples of problem-solving using divide-and-conquer, backtracking, and greedy-algorithm. (SQ)

316. Principles of Programming Languages. 3 lec. hr.; 3 cr. Prereq.: Computer Science 220, 240, 313, and 320. Principles and implementation of programming languages. Topics include: the procedural, object-oriented, functional, and logic programming paradigms; syntax (BNF, expression grammars, operator precedence and associativity); variables (scope, storage bindings, and lifetime); data types; control structures; function call and return (activation records and parameter passing); formal semantics. Programming assignments.

317. Compilers. 3 hr.; 3 cr. Prereq.: Computer Science 316. Formal definitions of programming languages: introduction to compiler construction including lexical, syntactic, and semantic analysis, code generation, and optimization. Students will implement portions of a compiler for some structured language. (SQ)

320. Theory of Computation. 3 hr.; 3 cr. Prereq.: Computer Science 111 and 220. Finite state machines, regular languages, regular expressions, grammars, context-free languages, pushdown automata, Turing machines, recursive sets, recursively enumerable sets, reductions, Halting problem, diagonalization.

323. Design and Analysis of Algorithms. 3 hr.; 3 cr. Prereq.: Computer Science 220 and 313. Advanced data structures: B-trees, graphs, hash-tables. Problem-solving strategies including divide-and-conquer, backtracking, dynamic programming, and greedy algorithms. Advanced graph algorithms. Time complexity analysis. NP-complete problems. Applications to sorting, searching, strings, graphs. Programming projects. (SQ)

331. Database Systems. 3 hr.; 3 cr. Prereq.: Computer Science 220, 313. ER modeling; functional dependencies and relational design; file organization and indexing; relational algebra and calculi as query languages; SQL; transactions, concurrency and recovery; query processing. Programming projects.

332. Object-Oriented Databases. 3 hr.; 3 cr. Prereq.: Computer Science 331. Review of basic database components and architecture; comparisons of OO databases with relational databases; modeling languages and methods, data definition languages; schema design methodology; the role of inheritance, object identity, and object sharing in OODBs; file structures and indexes for OODBs; transaction processing; concurrency control and recovery; development of database applications using a commercial OODB system.

334. Data Mining and Warehousing. 3 hr.; 3 cr. Prereq.: Math 241 and Computer Science 313. Data mining and data warehousing: data warehouse basics; concept of patterns and visualization; information theory; information and statistics linkage; temporal-spatial data; change point detection; statistical association patterns; pattern inference and model discovery; Bayesian networks; pattern ordering inference; selected case study.

335. Information Organization and Retrieval. 3 hr.; 3 cr. Prereq.: Computer Science 331. Concepts of information retrieval: keywords and Boolean retrieval; text processing, automatic indexing, term weighting, similarity measures; retrieval models: vector model, probabilistic model; extended Boolean systems: fuzzy set, p-norm models; linguistic model; extensions and AI techniques: learning and relevance feedback; term dependence; document and term clustering; network approaches; linguistic analysis and knowledge representation. Implementation: inverted files; efficiency issues for large-scale systems; integrating database and information retrieval.

340. Operating Systems Principles. 3 hr.; 3 cr. Prereq.: Computer Science 220, 240, and 313. Principles of the design and implementation of operating systems. Concurrency, multithreading, synchronization, CPU scheduling, interrupt handling, deadlocks, memory management, secondary storage management, file systems. Programming projects to illustrate portions of an operating system. (SQ)

342. Operating-System Programming. 3 hr.; 3 cr. A study of the internal structures of a particular operating system such as Unix, or another chosen by the department. (The operating system to be studied is announced at registration time.) Projects are assigned which involve system calls, use of the I/O and file systems, memory management, and process communication and scheduling. Projects may also involve developing new or replacement modules for the operating system Such as the command interpreter or a device driver. A student may receive credit for this course only once. (SQ)

343. Computer Architecture. 3 hr.; 3 cr. Prereq.: Computer Science 240. Instruction set architectures, including RISC, CISC, stack, and VLIW architectures. The memory hierarchy, including cache design and performance issues, shared memory organizations, and bus structures. Models of parallel computing, including multiprocessors, multicomputers, multivector, SIMD, PRAM, and MIMD architectures. Pipelining models, including clocking and timing, instruction pipeline design, arithmetic pipeline design, and superscalar pipelining. (SQ)

344. Distributed Systems. 3 lec., 1 lab. hr.; 3 cr. Prereq.: Computer Science 340. Issues in the implementation of computer systems using multiple processors linked through a communication network. Communication in distributed systems including layered protocols and the client-server model; synchronization of distributed processes and process threads.

345. Logic Design Lab. 2 lec., 3 lab. hr.; 3 cr. Prereq.: Computer Science 340. Design principles and laboratory implementation of logical devices from flip-flops to peripheral interfaces.

348. Data Communications. 3 hr.; 3 cr. Prereq.: Computer Science 343. Computer communications and networks; carriers, media, interfaces (RS 232, RS 422, CCITT); circuit types, data codes, synchronous and asynchronous transmission; protocols (OSI, TCP/IP); modems, multiplexors, and other network hardware; error correction and encryption; voice and data switching: local area networks, ISDN, packet switching; issues in the architecture, design, and management of networks. (SQ)

352. Cryptography. 3 hr.; 3 cr. Prereq.: Computer Science 313. An introduction to cryptographic practices, concepts and protocols. Topics include the mathematical foundations for cryptography, public key methods (e.g., RSA and El Gamal), block ciphers (e.g., DES and Rijndael), key agreement architectures (Diffie-Hellman), linear feedback shift registers and stream ciphers (e.g., A5 for GSM encryption), signatures and hash functions, (pseudo) random number generators and how to break the ENIGMA machine.

355. Internet and Web Technologies. 3 hr.; 3 cr. Prereq.: Computer Science 313. Internet protocol stack, analysis of representative protocols; Internet applications: client-server architecture, popular Internet application protocols, Internet application design, client side programming, server side programming, Web application and website design; programming projects.

361. Numerical Methods. 3 hr.; 3 cr. Prereq.: Computer Science 211 and Math 201. Numerical methods and efficient computation, approximation, and interpolation. Computer solution of systems of algebraic and ordinary differential equations.

363. Artificial Intelligence. 3 hr.; 3 cr. Prereq.: Computer Science 316. Principles of artificial intelligence. Topics include logic and deduction; resolution theorem proving; space search and game playing; language parsing; image understanding; machine learning and expert systems. Programming projects in LISP, PROLOG, or related languages. (SQ)

368. Computer Graphics. 3 hr.; 3 cr. Prereq.: Computer Science 220 and 313. Introduction to the hardware and software components of graphics systems, representations of 2D and 3D primitives, geometric and viewing transformations, techniques for interaction, color models and shading methods, algorithms for clipping, hidden surface removal, and scan-conversion. Programming projects using a graphics API to demonstrate the process of computerized image synthesis. (SQ)

370. Software Engineering. 4 lec., 1 lab. hr.; 3 cr. Prereq.: Computer Science 220 and 313. Principles of software engineering including the software life cycle, reliability, maintenance, requirements and specifications, design, implementation and testing. Oral and written presentations of the software design. Implementation of a large programming project using currently-available software engineering tools.

381. Special Topics in Computer Science. 381.1-381.4, 1-4 hr.; 1-4 cr. Prereq.: Permission of department. No more than 3 credits of CS 390-397 may be used as an elective for the Computer Science major or minor.

390. Honors Readings in Computer Science. 3 hr.; 3 cr. Prereq.: Junior or senior standing and permission of instructor. Students will study and report on survey and research papers dealing with various current topics in computer science selected by the instructor. Topics for each offering of the course will be announced at registration time.

391. Honors Problems in Computer Science. 391.1-391.3, 1-3 hr.; 1-3 cr. Prereq.: Permission of department. Open to students majoring in computer science who, in the opinion of the department, are capable of carrying out the work of the course. Each student works on a research problem under the supervision of a member of the staff.

393. Honors Thesis. 3 hr.; 3 cr. Prereq.: Junior or senior standing and approval of the Department's Honors and Awards Committee. The student will engage in significant research under the supervision of a faculty mentor and a thesis committee consisting of the mentor and two additional faculty members. The thesis proposal and committee must be approved by the Departmental Honors and Awards Committee. Upon completion of the research paper, an oral presentation of the results, open to the public, will be given. With the approval of the mentor, thesis committee, and the Department's Honors and Awards Committee, the course may be repeated once for credit when the level of the student's work warrants a full year of effort.

395. Research Projects. 395.1-395.3, 1-3 hr.; 1-3 cr. Prereq.: Permission of department. Open to majors and non-majors who, in the opinion of the department, are capable of carrying out an independent project of mutual interest under the supervision of a member of the staff.

398. Internship. 398.1, 45 hr.; 1 cr.; 398.2, 90 hr., 2 cr.; 398.3, 135 hr.; 3 cr. Prereq.: Completion of 15 credits in computer science and departmental approval. Computer science students are given an opportunity to work and learn for credit. Students should consult the College's Office of Career Development and Internships for listings of available internships and procedures for applying. A proposal must be approved by the department before registration. The student's grade will be based on both the employer's and faculty sponsor's evaluations of the student's performance, based on midterm and final reports. A limit of 6 credits of internships may be taken. Computer Science 398 may not be applied to the computer science major or minor.