COMPUTER SCIENCES AT PURDUE UNIVERSITY 1962 to 2000

JOHN R. RICE AND SAUL ROSEN

Abstract

Purdue University established the first academic department of computer sciences in 1962. The events, starting in 1959, leading to its establishment are chronicled, and then its first 35 years of development are presented. There are substantial data available about the department's evolution, some of which are presented here. There was a steady increase in the size and activities of computer science departments nationally, which were reflected in Purdue's experience. The two periods of crises in the department's history were triggered by the two periods of national enrollment explosion in the number of undergraduate majors.

Authors Biography

Saul Rosen was a computing pioneer, who, in the words of his 1984 ACM Distinguished Service Award, was always dedicated to extending the frontiers of Computer Sciences. He was born on February 8, 1922 at Port Chester, NY and graduated from City College of New York in 1941. He received his Ph.D. in mathematics from the University of Pennsylvania in 1954. After two years on the faculty at Wayne (now Wayne State) University he was employed by Burroughs Corporation (1956−1958), Philco Corporation (1958−1961) and as an independent consultant. He finally found his true home on the faculty of Purdue University where he served for 28 years (1963−1991). He was particularly active in the publication work of the Association for Computing Machinery (first Managing Editor of the Communications of the ACM) and in the founding of the Annals, of which he was an editor for the last 12 years of his life. The book, Programming Systems and Languages, which he edited in 1967, is a benchmark for historians of programming. He died on June 9, 1991 in West Lafayette, IN. He is listed as an author because of his major contributions to developing the material of this history.

John Rice is a computing pioneer who, in the words of the citation for his election to the National Academy of Engineering, was the founder of the field of mathematical software. He was born on June 6, 1934 in Tulsa, OK and graduated from Oklahoma State University in 1954. He received his Ph.D. in mathematics from the California Institute of Technology in 1959. After a year as a National Research Council post-doc, he joined the General Motors Research Laboratories for four years. He found his true home on the faculty of Purdue University where he has been for 38 years. He was active in the publication work of the Association for Computing Machinery as founder and Editor-in-Chief of the first ACM Transactions (Mathematical Software: 1975−1993), and as a member of the Publication Board for six years. He has authored over 20 books, mostly in computer science, between 1964 and 2000.

1. Formation of the First CS Department

An ad hoc committee on computing was created at Purdue in 1959. On November 3, 1959, Harold DeGroff (Head of Aeronautical Engineering), wrote George Hawkins, (Dean of Engineering), to recommend the creation of a computer laboratory with "at least two associate professors, three assistant professors, eight to 10 instructors…; the central idea here, of course, is to establish a computer research center with an emphasis on graduate work in this field". Faculty costs were estimated at $100,000 annually. This committee continued to discuss the possibilities and on March 28, 1961, Stanley Reiter, (Professor of Industrial Management), summarized the committee's view in "Proposal of ad hoc Committee on Computers to President Hovde". This recommends a Computer Sciences Center responsible for both research computing and computer science education. It states there would be: "(A) a strong professional group in numerical analysis and applied mathematics; (B) a similar group in mathematical logic and advanced computing equipment; (C) a graduate program in Computer Sciences".

From 1956−1961, there was serious conflict between the mathematics department and William L. Ayres, Dean of Sciences, Education and Humanities, (SEH). In 1959, Carl Kossack, (Head of Mathematics), submitted proposals to President Hovde to establish a School or Division of Mathematical Sciences, independent of SEH, and with a number of sub-departments. On March 9, 1959, he invited the President to a meeting to develop "a long range and comprehensive plan in looking toward the introduction of computing sciences into the several curricula of the university". There is no record of specific actions coming from this meeting but President Hovde was informed about the idea thereafter. Kossack left Purdue shortly thereafter, due to the conflict with Dean Ayers.

President Hovde finally addressed the conflict in early 1961 by moving the Mathematics Department to the School of Engineering, greatly increasing its budget, and renaming it the Division of Mathematical Sciences. A March 31, 1961 memo by Virgil L. Anderson, Director of the Statistical and Computing Laboratory, proposes to organize the division into four departments: Mathematics, Statistics, Computer Sciences, and Mathematics Education. Paul Chenea (former Head of Mechanical Engineering), was appointed temporary Head of the new Division of Mathematical Sciences. In the summer of 1961, Chenea became Academic Vice President and Dean Hawkins served as temporary Head of Mathematical Sciences. In the summer of 1961, Felix Haas, (Head of Mathematics at Wayne State University), became Head of Mathematical Sciences. In February 1962, the Division of Mathematical Sciences was given control of the Computer Sciences Center (a computing service organization) in the School of Engineering. It was agreed by Haas, Hawkins, and Hovde that the division would be organized into three academic departments: Mathematics, Statistics, and Computer Sciences, plus a Center for Computing Services.

A search was started immediately for a department head. Once Bill Miller, (Head of the Division of Applied Mathematics at Argonne National Labs), removed himself from consideration, the position was offered to Samuel D. Conte, (Manager of Programming and Analysis at Aerospace Corporation). Conte has been a professor at Wayne State University along with Haas from 1954−1956. On October 24, 1962, President Hovde asked for and received approval from the board of trustees to change "the internal administrative organization of the Division of Mathematical Sciences…effective October 1, 1962". The Department of Computer Sciences and the Computer Sciences Center were listed as components of the division, along with the Departments of Mathematics and Statistics, and a statistical laboratory. Professor Conte was listed as Chairman of the Department of Computer Sciences and Director of the Computer Sciences Center. The October

24 entry of the Board of Trustee's minutes makes it very clear that the Purdue Department of Computer Sciences was officially established in the fall of 1962.

During this period, the SEH School was reorganized and a School of Sciences created with Haas as Dean. The Division of Mathematical Sciences was moved to this school, along with Biology, Chemistry, and Physics. The turmoil created by the conflicts in SEH made it easier to create a new department and reorganize the structure of mathematics. It is not clear to what extent the report of the ad hoc committee on computing, or Anderson's memo influenced Haas' thinking about the future status of computing and Computer Sciences at Purdue. Haas recalls a meeting with Hawkins and Hovde, probably before he officially started at Purdue, in which they agreed that the Division of Mathematical Sciences would be internally divided into three academic departments: Mathematics, Statistics, and Computer Sciences, and a Computer Sciences Center to provide computing services to the whole university. According to Haas, the fact that computing and Computer Sciences were to be among his responsibilities at Purdue, made the position as Head of the division more attractive to him.

When Conte arrived at Purdue in the summer of 1962, he faced two major challenges. One was to raise computing services at Purdue to an appropriate level for a major research university. The other was to organize a new Computer Sciences department, for which no model then existed anywhere. A great deal of his energy was devoted to computing services; this activity is described in detail in [Rosen and Rice, 1994].

2. Establishing the Department of Computer Sciences

The department started in 1962 with three young professors already at Purdue: Richard Kenyon, Robert Korfhage and L. Duane Pyle. Both Kenyon and Pyle had received Ph.D.s from Purdue in 1960, (Kenyon in electrical engineering and Pyle in mathematics), and stayed on to work in the computer services center. Korfhage had a Ph.D. in Information Sciences from Michigan and came to Purdue in Mathematics. These three, plus Conte, formed the initial professorial faculty in Computer Sciences.

After Conte, the first faculty member hired for the new Computer Sciences department was Saul Rosen. Conte had known Rosen at Wayne State University before they both left the university in 1956. Rosen worked in the software area for Burroughs and Philco Corporation and then as an independent consultant. He contacted Conte about possible consulting work on the West coast, and Conte suggested Rosen join the new Computer Sciences department that Conte was forming at Purdue. He arrived at Purdue in early 1963. The Computer Sciences department started out with a number of graduate students, several of whom who had come to Purdue from Aerospace Corporation along with Conte. From the beginning, the department recognized three major areas: numerical analysis, systems, and theory. Conte taught the first course in numerical analysis, Rosen taught the first course in computing and programming systems, and Korfhage taught the first course in algorithms and automata.

There are four natural phases to the development of the department. In the 1960s, the effort was to define courses, degree programs, and indirectly the field itself. The 1970s saw the department's maturation and growth into a typical university department. The 1980's started with a series of crises, some nationwide, and some internal to Purdue, which eventually gave the department a considerably different character than it had in the 1970s. The 1990s started with a well established department, but two more crises occurred in the mid 1990s, one national and another internal to Purdue. The rest of the paper is organized around these periods. Table 1 presents a chronology of the principal events and milestones for 1962–2000.

Table 1: Milestones of the Computer Sciences Department at Purdue, 1962–2000

1962	Department formation, M.S. and Ph.D. programs started
1964	First M.S. degrees awarded (3)
1966	First Ph.D. degrees awarded (2)
1967	Move to Mathematical Sciences building
1968	Undergraduate program started and first B.S. degrees awarded
	One hundredth M.S. degree awarded
	Regular faculty size reaches 10
	Department and Computing Center completely separated
1972	Regular faculty size reaches 20
1977	Five hundredth M.S. degree awarded
1978	Five hundredth B.S. degree awarded
	Department acquires first computer facility (VAX 11/780)
1979	Conte retires as Head: Denning appointed
1981	Crisis: Enrollment explosion arrives
1982	One hundredth Ph.D. awarded
1983	Large loss of key faculty
	Denning leaves; Rice appointed Head
1984	Crisis: No space for faculty, students, laboratories, or staff
1985	Move to new building
1987	Regular faculty size reaches 31
	One thousandth M.S. degree awarded
1989	Two thousandth B.S. degree awarded
1994	Two hundredth Ph.D. degree awarded
1996	Regular faculty size drops to 26
	Crisis: Second enrollment explosion arrives
	Crisis: No laboratory space for teaching or research
	Rice resigns as Head
1997	Sameh appointed Head
2000	Three thousandth B.S. degree awarded

3. The 1960s: Establishing the Curriculum

The first task of Conte was to hire some faculty and define a graduate program. The course offerings planned were to be large enough graduate courses for the M.S. and Ph.D. degrees plus undergraduate service courses in programming. Descriptions of these initial courses are given in Appendix 2. By the end of the year there were seven teaching faculty, including Conte, Kenyon, Pyle, Korphage, and Rosen. Also teaching were Don Novotny, an industrial engineering Ph.D. candidate and full-time instructor, and Rosemary Stemmler, a full-time instructor. Although not all the faculty taught full time, they could offer over 20 courses a year, which was ample to support the planned program.

The Computer Sciences department struggled to establish its identity during those first few years. The Division of Mathematical Sciences controlled requirements for the Ph.D. degree, and mathematicians in the division felt that no one should have a Ph.D. from the division without having shown mastery of important parts of the mathematics curriculum. The qualifying examination for the mathematics Ph.D. was designed to eliminate students who did not show promise as researchers in mathematics, and Computer Sciences Ph.D. candidates were expected to pass some of these same qualifying exams. One might argue for or against the merits of such policies, but the net result was that a number of promising students, especially in the systems area, were discouraged from trying to obtain the Ph.D. degree in Computer Sciences. It took several years before the department was able to control its own requirements for advanced degrees.

In 1963 there were three new faculty members: Richard Buchi (theory), Walter Gautschi (numerical analysis), and John Steele (programming systems), who worked primarily in the Computer Sciences Center and rarely taught. The following year John Rice was hired in numerical analysis. No new faculty was hired in 1965, and only one, Carl de Boor in numerical analysis, was hired in 1966. de Boor was the first of a number of young Ph.D.s hired who became influential members of the department. Robert Lynch in numerical analysis and Paul Young in theory were hired in 1967, Jay Nunamaker in business applications was hired in 1968, and Victor Schneider and Vincent Shen, both in systems, were hired in 1969. Also hired during this period was Maurice Halstead, a senior person in programming systems and software engineering.

The new Mathematical Sciences building was completed in 1967 and the department, along with Statistics, moved there. The Computer Sciences Center occupied the two floors below ground. The department occupied the fourth floor, which was substantially larger than the previous space and also much nicer. In the beginning, space was so ample that some graduate students were given faculty offices (with windows!). Fifteen years later, even with space on several other floors, people were jammed together.

In 1968, there was a major management change. Conte had been both Director of the Computer Sciences Center, a services organization, and Head of the department. Rosen was appointed Director of the Computer Sciences Center, which was renamed the Purdue University Computing Center (PUCC). Very close ties were established between the computing center and the department while Conte was Head of both; this friendly cooperation continued under Rosen's, and later Steele's, direction and still persists today.

Initially the three departments were only partially independent within the Division of the Mathematical Sciences. They set degree requirements separately, but there was only one graduate committee and one Ph.D. qualifying exam system. The three departments had separate personnel committees, but not separate budgets. This arrangement was appropriate in view of the small sizes of the Computer Sciences and Statistics departments, and the administrative skill of Haas. This arrangement was, of course, also a continual source of friction, and the three departments gradually became truly independent. The final step was the complete separation of budgets in 1969.

3.1 Graduate Program

The M.S. degree was designed as a program to train computer scientists for industry. Students with B.S. degrees in other fields (of course there were no B.S. graduates in Computer Sciences at the time), were given a broad introduction to numerical methods, programming systems, and theoretical computer science. Ten graduate courses were required for the M.S. degree, with wide flexibility given in selecting them. The principle requirement was that one course must be taken from each of the three main areas, theory, numerical analysis and programming. Students often took a few related courses from engineering, applied mathematics, or statistics that are not in the list of Appendix 1. The first three M.S. degrees were awarded in the spring of 1964. The number of M.S. graduates per year rose rapidly thereafter as seen in Table 1.

Defining the Ph.D. was not difficult in the areas of numerical analysis and theory, since there were already well-established research sub-disciplines in these areas. Furthermore, the qualifying exam system was reasonably compatible with these two areas. These exams were uniform and naturally very mathematical in nature. Defining the Ph.D. in programming systems was no so simple. Most of the research was in industry; there were no standard research journals; many important ideas and results were published in ad hoc ways – or even not at all. There were no textbooks and very few research monographs; defining course work and evaluating theses were difficult. The qualifying exam was a particular challenge for students in this area. The "standard body of knowledge" for this exam was missing, yet the existing mathematics exams (even the one in applied mathematics) were far removed from students' needs and interest. Students were asked to become expert in these outside areas, which they viewed as both a very difficult task and a waste of time.

The first two Ph.D. graduates were in 1966: Karl Usow, a student of Rice, and Kenneth Brown, a student of Conte. The following year there were five Ph.D. graduates. The first Ph.D.s in programming systems were not until 1969: Larry Axsom and Edouard Desautels, both students of Rosen. It is not always easy to decide whether some Ph.D. students in the early days were in Mathematics or Computer Sciences. All senior Computer Sciences faculty also had appointments in Mathematics; the qualifying exams were the same, and there was a single graduate committee. The Ph.D. requirements in the Division of Mathematical Sciences, unlike those for the M.S., were the same for all departments, and Ph.D. degrees are not officially labeled by department. Thus, there were Ph.D.s in Computer Sciences whose advisors were not on the Computer Sciences faculty, and several Computer Sciences faculty (for example, Buchi, Gautschi, Lynch, Rice, and Young) had Ph.D. students in Mathematics.

3.2 Undergraduate Program

The undergraduate program evolved initially from very sparse course offerings in programming, to a Computer Sciences option in the mathematics department, to a separate B.S. degree in Computer Sciences, approved in 1967. Conte was an active member of the Association for Computing Machinery committee that studied and recommended a model B.S. degree program. The result, known as Curriculum'68, was very close to the degree program at Purdue, which was one of the test-beds for developing Curriculum'68. B.S. degrees were awarded immediately after the degree was approved, because many students could and did transfer from the Computer Sciences option in Mathematics and met the new degree requirements within a year.

The similarities between these B.S. degree curricula are illustrated in Table 2, which compares course requirements for (1) the B.S. degree in Computer Sciences, (2) the B.S. degree in Mathematics within the Computer Sciences option, and (3) the model Curriculum'68. The principal difference between the Computer Sciences major and Curriculum'68 was the increased emphasis on a theory at Purdue and that programming languages was covered in several courses by Curriculum'68. Descriptions of all the initial undergraduate Computer Science courses are given in Appendix 2. The Computer Sciences option in Mathematics simply had lower requirements, consistent with the requirements of the other Mathematics options.

Table 2: Course requirements for a B.S. degree in Computer Sciences, in Mathematics with a Computer Sciences opti and the Curriculum'68 model program.

Course	Computer Sciences Major	Computer Sciences Option in Mathematics	Curriculum'68
Calculus	3	3	3
Advanced calculus	1	1	1
Linear algebra	1	1	1
Programming 1 and 2	2	2	2
Numerical methods	1	1	1
Theory	3	2	1
Computer systems	2	0	2
Programming languages	0	0	1
Electives—Computer Sciences	2	2	2
Statistics	1	0	0
Electives—Mathematics/Computer Sciences/Statistics	0	0	2
Total courses	**16**	**12**	**16**

The B.S. degree was started in 1967, without a full range of appropriate undergraduate courses. The B.S. program relied heavily on graduate courses, and a typical B.S. degree included three to five courses at the dual graduate/undergraduate level. This situation reflected two facts. First the faculty was still not large enough to offer a full range of courses for the B.S., M.S., and Ph.D. degrees and a substantial service program, and second, material appropriate for undergraduates had to be offered in graduate courses, because entering graduate students rarely had a B.S. in Computer Sciences. There was a steady migration of material from the graduate level downward into the undergraduate courses as soon as faculty size and student backgrounds allowed it. It was not until well into the 1980s that the undergraduate Computer Sciences program included the variety of offerings that is common in the other sciences.

The most important fact about the Computer Sciences curriculum in these formative years was that the rapid evolution of the field rendered courses at all levels out-of-date in a few years. In the early days we hoped soon to be able to define courses Computer Sciences 101, 102, and 103 that would become semi-permanent fixtures analogous to Mathematics 101−103 or Physics 101−103. This still has not yet happened as of 2002 and it does not seem likely to happen soon. Some of the other lessons learned early were

It is unrealistic to teach programming to a mixed class of sciences, engineering, and business students.

It is difficult to keep students, teaching assistants, and even some faculty focused on the principles of computer science instead of the mechanics.

There is never enough money to provide the level of computing facilities that students deserve.

Programming assignments create new possibilities for student cheating; as soon as one cheating technique is counteracted, another is invented.

4. The 1970's: Maturation of the Department

By the beginning of the 1970s, the department had completed its pioneering years. Degree programs were established, there was a faculty of 15, and dozens of Computer Sciences departments at other universities, and Purdue's department was fully independent. The 1970s were to be a decade of consolidation and maturation. But there were still serious challenges; perhaps the most difficult was hiring faculty. By 1970, there were a significant number of Computer Sciences Ph.D.s, but it did not come close to meeting the demand. Computer Sciences departments were being rapidly established, the computing industry was expanding steadily, and other industries (oil, aerospace, and banking) began to hire significant members of Ph.D.s. Throughout the 1970s, almost every Computer Sciences department had unfilled faculty positions, as did many major industries. The increase in national Ph.D. production was slow, almost zero, in the latter part of the decade.

The regular faculty at Purdue increased from 15 in 1970−71 to 22 in 1979−80, the result of relentless recruiting. Young faculty, who were hired and later became important in the department include: 1972: Peter Denning, Michael Machtey, and Herbert Schwetman; 1976: Douglas Comer, and Christoph Hoffmann; 1977: Michael O'Donnell, and 1978: Buster Dunsmore. These gains were offset by losses of key faculty: Four members went to other positions – Pyle, de Boor, Nunamaker, and Schneider – and Halstead and Machtey died in 1979.

The faculty shortage was compounded by another trend that became widespread in the 1970s: the change from a mathematics-like discipline (using only paper and punched cards), to sciences-like discipline with a significant experimental sciences component. By the end of the 1970's, most departments, including Purdue's, had a significant experimental component. As the discipline moved in this direction, it adopted some of the practices of the experimental sciences. In particular, teaching loads had to be reduced to compensate for the increased effort needed to operate teaching laboratories and experimental research facilities. The fierce competition for faculty of course accelerated the reduction of teaching loads and the offers of equipment to attract faculty.

A significant effect of the lack of faculty was the heavy reliance on graduate teaching assistants. Although the department recognized that it was educationally unsound, graduate students sometimes taught other graduate students in the 1960s, and graduate students commonly taught upper division courses in the 1970s. There seemed to be no alternative except not to offer the courses.

A second serious challenge for Computer Sciences departments everywhere in the 1970 was to establish their scientific respectability. Many sciences and engineering faculties knew about computing only through contact with Fortran programming, and they assumed that was all there was to Computer Sciences. Even though the Purdue Department of Computer Sciences was consistently rated in the top ten[1] (higher than any other Purdue science department), it had to reaffirm its performance and value continually to other parts of the university. While there is still a residual of these feelings even today, by the end of the 1970s, the majority of the university's administration and faculty believed Computer Sciences was a serious scientific discipline that was here to stay.

The third serious challenge was the evolution of courses. In spite of repeated reorganizations of courses and the expansion of offerings, it seemed there was always some course that required

[1] These rankings are based on surveys made by the computer science profession (e.g., Computer Science Board) and not by more traditional academic ranking organizations. Computer Science was not yet recognized enough to be included in normal academic ranking procedures.

complete restructuring. The department simply did not have enough faculty to keep all the courses up-to-date at all times; this situation persisted into the 2000s.

The educational programs were fairly stable in size during the 1970s. From 1970−1979, the number of Ph.D.s awarded annually remained at six, and the number of M.S. degrees increased from 53 to 54. The number of B.S. degrees awarded annually grew from 32 to 92, but 71 had already been awarded in 1973−74.

The quality of the degree programs improved significantly during this decade. By the end of the decade all entering graduate students were required to have the equivalent of a B.S. in Computer Sciences. At the undergraduate level, the number of courses offered increased significantly, and better textbooks became available.

The decade ended with Conte being succeeded by Denning as Department Head in 1979. In Conte's 17 years as Head, he had guided the department from its pioneering infancy to a strong department, both nationally and within the university, which was a major achievement. The department also benefited greatly from the foresight of Haas, who became Dean of Sciences soon after the department was formed and later Provost. Already in the early 1960s, Haas foresaw that Computer Sciences would become one of the major scientific disciplines, and he supported Conte's efforts to keep Purdue's department growing and to become one of the best.

5. The 1980s: Decade of Crisis

The growth and maturation of the 1970s held the seeds for the crises that hit in the first half of the 1980s. These crises stemmed from the numerous needs and the lack of resources to meet them.

5.1 Crisis 1: Student Enrollment Explosion

The number of entering Computer Sciences freshmen grew gradually during the 1970s. About 80 to 100 entered from 1970−1974. Enrollment then increased to 150 a year from 1975−1977. In 1978 and 1979, the number increased to 200, then 300, and the crisis was on us (see Figure 3). This growth was nation-wide. For one year in the early 1980s, one national survey reported in the press stated that 9% of the high school graduates wanted to study Computer Sciences. If this percentage had continued, Computer Sciences would have had as many students as all of engineering! By the fall of 1981, there were over 500 freshmen starting out in Computer Sciences. Note that Purdue is unusual in that essentially all freshmen declare a major. Earlier groups of students were advancing through the curriculum, and the undergraduate courses overflowed, new sections were added, and then they overflowed again. The national nature of this explosion is discussed by [Denning, 1980] and [Margarrel, 1981]

Purdue's administration was very reluctant to limit the entering freshmen class in Computer Sciences, since there was strong pressure to increase the size of the Purdue student body. In 1982, it was agreed on to limit the freshmen class in Computer Sciences: higher SAT scores and class rank were required. The official number of freshmen majors dropped to about 350 in 1982 and stayed there until 1985, but even that number was beyond the capacity of the department.

The faculty soon realized that the administration had quietly created a new category of students, the pre-computer sciences classification. Essentially all the students who met the School of Sciences requirements, but not the Computer Sciences requirements, were put in this classification.

Thus there was no reduction in students in Computer Science courses, even though the number of official majors decreased. Pre-computer science students who made acceptable grades became official Computer Science majors in their third year, when they became upper division students. This action helped precipitate the 1983 crisis.

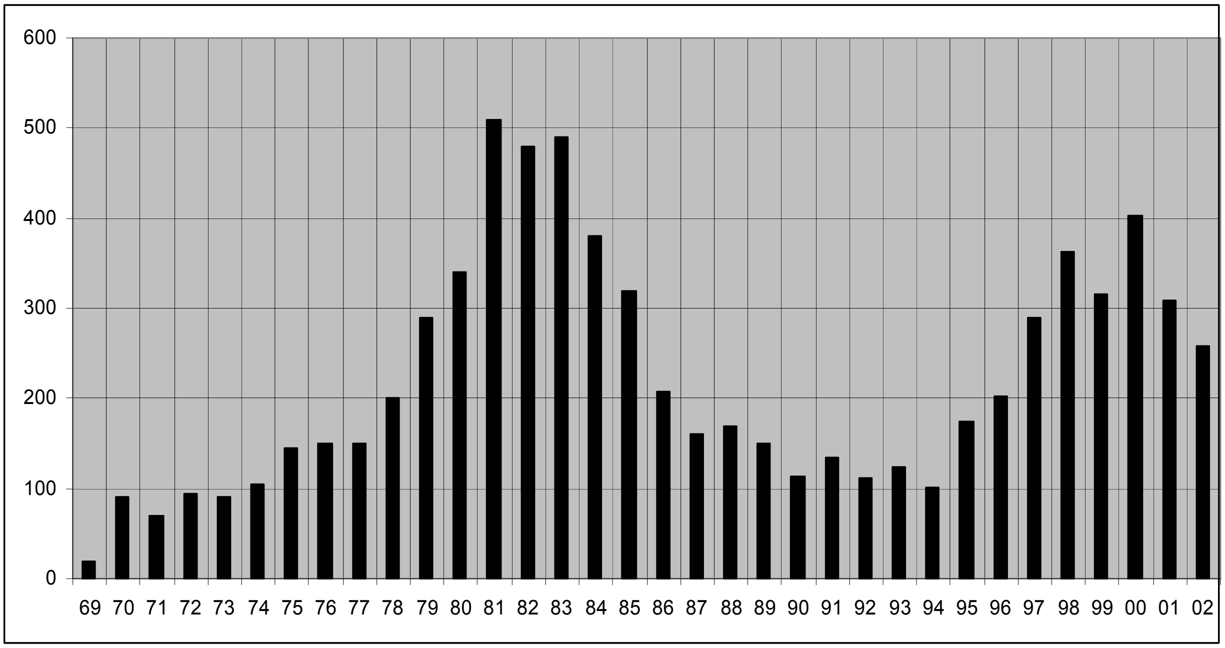

Figure 3. Entering freshmen who declared a Computer Sciences major from 1969–2002. For 1982–1984 and after 1998, the classification of pre-Computer Sciences major is included in the officially declared majors.

This administration did offer to increase the number of positions in the department, but that was a risk free offer. The department already had unfilled positions and having more of them would not cost the administration anything nor increase the number of faculty. The explosion was handled by increasing class sizes. Examples of the extreme situation during that period were: (1) The senior level course in numerical analysis was taught in a single lecture section with about 150 students and half-time teaching assistant grader; (2) The first-year graduate course in compilers had over 80 students and no teaching assistant grader; (3) Teaching assistants had 160 to 180 students in lower division courses.

There was a corresponding lack of computing facilities to support the courses. Examples of the poor service of the time sharing systems include: (1) Terminals reserved for "quick fixes'' were limited to 10-minutes of use and service was so slow that one could not log on in 10 minutes; (2) Terminals would automatically disconnect because they did not receive an echo of characters sent to the computer within 15 seconds and assumed the computer was down, rather than merely swamped.

5.2 Crisis 2: The Space Crunch

The growth of the 1970s had gradually taken up available space in the Mathematical Sciences building for all its occupants – Mathematics, Statistics, PUCC, School of Sciences administration, and Computer Sciences. When Computer Sciences started changing to an experimental, laboratory-oriented discipline in the 1980s, space was needed for departmental computing, teaching labs, and research labs. A few offices were converted into labs for research projects, but the department was severely constrained by the lack of space. The result was tighter and tighter packing, and most faculty simply could not engage in laboratory work. Five graduate students were placed in a 150-square-foot office, each with his/her own desk, chair, and some bookcase space. These offices were already over crowded when four students occupied them.

In 1983 a faculty member asked the department Head for a secretary. Heads usually reply that they do not have any money for that, but in this case, the department Head said, “I understand, let’s do it. Tell me where you want the secretary to be and I’ll hire one.” This offer was safe because the Head knew that there was no room in the department to add even one more desk.

5.3 Crisis 3: Establishing Departmental Computing Facilities

The department acquired its first general purpose computer in 1978, a VAX 11/780. It was the first VAX to be running VAX Unix outside the Unix developer sites (Berkeley and AT&T Bell Labs). It provided the faculty an interactive, time-shared computing environment. It was not practical for PUCC to provide this service on a widespread basis, and PUCC was unwilling to do so for just one department. It was, however, inevitable that the department would set up its own facilities as its needs became too specialized and too diverse to be satisfied by a centralized service center. This move was part of the nationwide trend of Computer Sciences to become more experimental and laboratory oriented.

This crisis was very real, but it was handled much more smoothly than the others because of the administration’s willingness to support this growth. The extent of the changes required is illustrated by the following data on budget changes from 1979−1989. The facilities operating budget went from $20,000 to $500,000, the staff went from 0 to 9, and the equipment value went from $250,000 to $3,500.00. In a 10-year period, a major new operation was established within the department. The operating budget was entirely from university funds; most of the equipment was purchased through government research grants.

While this crisis was handled smoothly, it did have its trying moments. The department, like many others, did not initially realize the necessity of a professional staff to operate the facilities. The early staff were regular faculty (D. Comer, then H. Schwetman) and students who took on this extra challenge. They did a superb job, but it was not their only job, and more than once the following occurred:

- Professor X has a paper that must go out today, and it is being revised on the computer.
- The computer crashes and Professor X rushes to the office of Professor Y, who is in charge of the facilities, to demand that the computer be brought up at once.
- Professor Y replies that she has two classes to teach, has not finished preparing for them, and may be able to work on the problem until late in the afternoon. All the students who might be able to bring up the system are in class, the library, or some place else.
- A heated discussion ensues between Professors X and Y about who is irresponsible, unreasonable, incompetent, and so forth.

These growing pains were on the whole minor, and the department did obtain excellent computing facilities, but such growth contributed to the space crisis. Fortunately computers were becoming smaller, or the crisis would have been even worse.

In response to crisis 3 in January 1984, a building was selected for renovation to house the Computer Sciences department. The renovation was completed quickly, and the building was occupied in the fall of 1985. The space was of excellent quality, and for a few years, the department enjoyed ample space. However, the need for labs, supporting staff, and research assistants grew rapidly, and by 1989, the packing process was being repeated.

5.4 Crisis 4: Loss of Key Faculty

The crises already discussed began in the early 1980s to create concern, and then alarm, among the faculty, who saw that to remain among the top ten departments, Purdue would have to make major investments to create the experimental sciences facilities needed. This meant more space for laboratories, more support for staff, and more computing facilities. Instead of addressing this challenge, many faculty felt the administration was letting it drown in a flood of students. The faculty realized that it was impossible to hire ten new professors, since they were not available. The faculty did hope, however, that the administration would help in other ways – more teaching assistants, more lab space, and better computing facilities. Faculty morale dropped steadily as it appeared that little help would be forthcoming.

This crisis should be placed in the context of the national situation: Enrollments were ballooning wherever they were not strictly limited; there was a national awareness that major investments were needed for experimental Computer Sciences facilities, see [Denning, 1980].and [Margarrel, 1981] Many universities responded with major programs in Computer Sciences, and the Purdue faculty felt that all the leading departments were receiving $15 million or $30 million for a new building and 20 new positions. The faculty felt Purdue was going to be left behind.

In the summer of 1981, a group (Denning, Rice, Larry Snyder, and Young) from Computer Sciences met with Provost Haas to discuss the situation. They had prepared a plan[2] to maintain the excellence of the department. The faculty's sense of unease was expressed, and the Provost responded by saying that Purdue strongly supported the department and would not let it fall apart. He noted that Purdue could not let people like those present become so unhappy that they left Purdue; it would ruin the department. A year-and-a-half later, Denning and Young had resigned, and a third (Snyder) resigned a year after that.

The plan presented was realistic, the faculty would receive less than it needed, and the administration would give more than it wanted. The plan was agreed to in principle, but not as an itemized list of commitments. The plan and resulting actions did not fully dispel the belief that Purdue was willing to settle for a second-tier Computer Sciences department. In fall 1983, the faculty was systematically surveyed about the department's problems and priorities. Of 22 items, the following were judged to have the highest priority (in the order listed).

1. Large classes
2. Few Ph.D. students
3. Lack of laboratory space
4. University's commitment to maintaining a top-tier department

Overall, faculty ratings of the department's and university's performance in attending to these items were, respectively, D+, C-, D, and B-.

In 1983 and 1984, the department lost ten of its faculty, including many of its key people. The departing faculty would have constituted one of the better departments in the country, and the loss was clearly a major blow to Purdue. The departmental crisis caused most of the departures. See Figure 4 for data on regular faculty size from 1963−1998. The departures and hiring in 1983 and 1984 are tabled below. The faculty hired in 1983-1984 did not adequately replace the departures.

Purdue Computer Science Faculty Departures and Hires, 1983 and 1984

Rank	Departures	Hires
Full Professor	P. Denning, R. Buchi, H. Schwetman, L. Snyder, P. Young	
Associate Professor	Dorothy Denning, Mike O'Donnell, Dennis Gannon	Alberto Apostolico, Dan Marinescu, Samuel Wagstaff
Assistant Professor	Fran Berman, Jan Cuny	Chandrajit Bajaj, Wayne Dyksen, Piyush Mehrotra, Rami Melhem, Walter Snyder

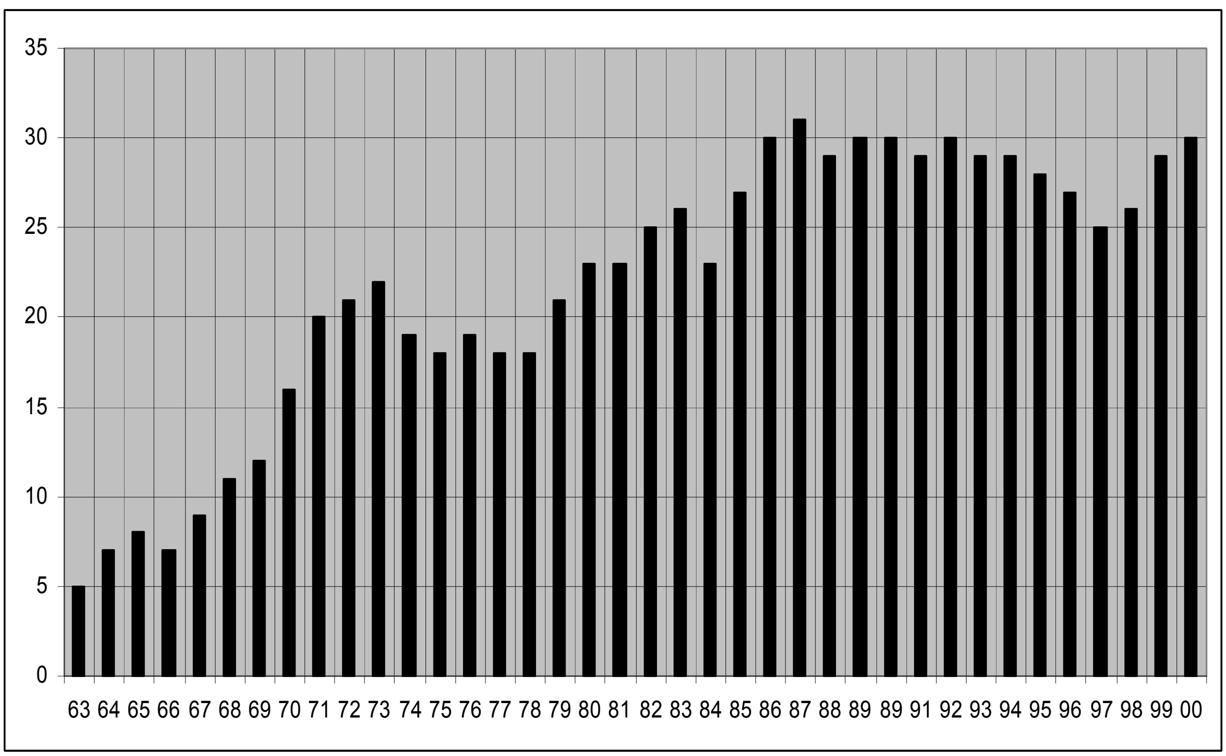

Figure 4. Regular faculty in the Department of Computer Sciences from 1963−1998.

The search for a replacement for Denning as Department Head was educational. Rice was appointed Acting Head, and a vigorous search for a new Head started. The dean outlined how the candidates would be winnowed out; the department opined that Purdue would be lucky to have any serious candidates to winnow. A year later, only three candidates had been found who were both interested and interesting enough to be interviewed. Two were offered the position and turned it down; the third said it would be a wasted effort to make an offer. After a year and a half, Rice agreed to become actual Head.

As the crisis deepened the administration became more convinced that it was real and efforts should be made to save the department. The entire Computer Sciences faculty was given a 1% mid-year raise as a sign of commitment from the university and it was decided to completely renovate the Memorial Gymnasium and make it into the Computer Sciences building.

5.5 Recovery

Recovery from these crises began in 1985. Moving into the newly renovated Computer Sciences building dramatically improved morale. This physical improvement was accompanied by initial solid evidence that the flood of students was receding. Only a little over 300 in the entering Purdue freshmen class had declared to be Computer Sciences majors. Even though severe damage was caused by some of these crises, there had been real success in meeting some of them. From 1980–1989, the department's budget increased from $1.1 million to $3 million, at least a million dollars more than the inflation rate.

The first priority after the faculty exodus in 1983–1984, was to rebuild the faculty. The nation-wide shortage of computer scientists made this a difficult challenge, but one that had to be met. New hiring really began in earnest in 1984 with six new regular faculty appointments. By 1989, the regular faculty had grown to 30, of which 18 were hired in 1983 or later. Not one of the 19 regular faculty hired in the 8-year period 1968–1975 remained; there were five "old timers", Conte, Gautschi, Lynch, Rice, and Rosen, and seven "middle-aged" faculty members. Most of the new faculty hired were, of course, young, even though losses had been heaviest among the more senior faculty. Five Associate Professors were hired, but the majority were relatively new to the rank. Three Full Professors were hired. Rao Kosaraju was appointed the Loveless Distinguished Professor of Computer Sciences in 1986, but he returned to Johns Hopkins after a year. The other Full Professors appointed were Richard DeMillo as Director of the Software Engineering Research Center (SERC), and Elias Houstis. The paper [Rice and Rosen, 1994] gives details of the hiring and departures of the faculty from 1963 through 1990.

A second high priority was to expand the experimental research activities now that space was available. By 1989, the department had 11 substantial research activities, ten with laboratories. The two largest projects were SERC and Computing About Physical Objects (CAPO) described below. The others were Computational Combinatorics, Cypress (networking) ELLPACK (scientific computing), Graphics, Interbase (databases), RAID (databases) Scientific Visualization, Shadow Editing (operating systems), and Xinu (operating systems). Growth in research in general and experimental research in particular, is illustrated by the increase in research funding from $447 thousand in 1980 to $2.6 million in 1989. There was also a substantial increase in the number of Ph.D. students.

The SERC is part of the National Sciences Foundation's Industry/University Cooperative Research program. It started jointly with the University of Florida and 15 industry affiliates. The

SERC was established at Purdue primarily through the efforts of Conte, and it became operational in 1985. DeMillo came as the permanent director, and substantial laboratory space and equipment was provided for SERC's use. By 1989, SERC involved 12 faculty and 14 graduate students at Purdue.

The CAPO project started in 1987 with support from the National Sciences Foundation's Coordinated Experimental Research program, plus other agencies and Purdue. This project originated in 1986 by Hoffmann, Houstis, and Rice; by 1989 the project involved seven faculty, three postdocs, and over 20 research assistants and staff personnel.

The new space acquired on moving into the Computer Sciences building also allowed the department to establish teaching laboratories. In the first year, 1985, there were four – two for the elementary personal-computer-based service course), one for the first course for Computer Sciences majors), and one for graduate courses in operating systems and networking. By 1989, this number had doubled. Operating laboratories are considerably more expensive in manpower than ordinary lecture courses, as they must have laboratory assistants and supervisors of various kinds. The funding for this expansion came primarily from the reduction in Computer Sciences majors; as fewer sections of various courses were needed; teaching assistants were reassigned to help in labs.

6. The 1990s: Recovery, Then More Crises

By the end of the 1980s, the department had recovered somewhat from the crises. There were long term effects, most notably in that its national ranking among Computer Sciences departments had dropped from the top ten to the high teens. Even so, the student pressures were gradually easing and the research activities were steadily expanding. Two ordinary events, one in 1992 and one in 1995, presaged more crises to come: In 1992 Harry Morrison was appointed Dean of Sciences and in 1995 the size of the entering freshman class started growing again.

6.1 The Second Student Enrollment Explosion

By 1995, it was clear that a second national explosion in Computer Sciences enrollments was underway. Purdue observed this earlier, because unlike most other universities, it has entering freshmen declare their major. If the 1996 entering class size just continued, the number of undergraduates in Computer Sciences would double or more. The department simply did not have the resources to handle these students. Indeed, the department had been in the process of reducing maximum class sizes to respectable levels, e.g., 40 students in upper division courses.

The problem was compounded by the fact that enrollment in other science departments were dropping or steady. The increase in Computer Sciences majors was needed by the School of Sciences in order for it to meet its goals for Purdue's expectations of slow, but steady, enrollment growth. Reluctantly, the Dean agreed to instituting higher admission standards for Computer Sciences freshmen. The target enrollment would be about 1000 majors in Computer Sciences; the number it had it the late1990s. However, the standards were again circumventing as they had been in the middle 1980s (a new student category "pre CS" was created and all students who wanted to study CS and didn't meet the higher standards were admitted in this category). Thus, Computer Sciences would handle 40% of the schools majors with only 15% of the school's faculty. In addition, Computer Sciences had a very large service course responsibility, as did some of the other science departments.

6.2 The Second Space Crunch

The space for growth acquired in 1984 was gradually absorbed by expanded research programs and computerized instructional laboratories. By 1990, the four of the five lecture rooms had been converted to labs. Only the dramatic decrease in the size of computers kept the computer room from over-flowing, but the facilities support staff expanded. The PUCC helped the department greatly by moving three of its instructional labs for service courses to other buildings. Still, the available space was exhausted by the early 1990s, and the packing of people and equipment started again.

The university administration believed the department's space problem should be solved by the School of Science, since some of its departments had large amounts of unused or marginally used space. All of the three departments from the now disbanded Division of Mathematical Sciences were squeezed tightly and needed new space, but only Computer Sciences was growing steadily and needed substantial amounts.

6.3 Crisis: Conflict over Resources

By 1993, it was clear to the department that substantial increases in resources (faculty, space, and support staff) were required. The university administration believed these should be obtained through reallocation within the School of Science. The new Dean of Science did not want to reallocate resources to Computer Science. He started withholding permission for the department to fill vacant positions and the size of the regular Computer Science faculty dropped from 30 in 1992 to 25 in 1997. This was, as usual, a difficult time to recruit Computer Sciences faculty, but those positions that Computer Sciences was allowed to fill, were filled with well qualified people.

At first, the new Dean did not consider allocating any space from other departments to Computer Sciences. As the problem grew worse, he told Physics in 1995 to provide Computer Sciences with some space. They had just established a 3000 ft^2 museum for high school students and had two rooms totaling over 1500 ft^2 for student lounges (Physics had just over 100 majors, while Computer Sciences had over 500 going on to 1000). The space offered was an abandoned underground laboratory with radiation shielding 3 feet thick and a very strange configuration. At the time, Physics had several large, reasonable, and empty labs.

The conflict over resources led to greater and greater friction between Dean Morrison and Rice (still Head of Computer Sciences). The situation became so tense that in April 1995, a large delegation of Computer Sciences faculty met with the Dean to express support for Rice as Head and for his efforts to obtain the resources the department needed. The Dean's response was to send an emissary to Rice the same day with an ultimatum: Agree within a week to resign as Head effective July 1, 1995 or be fired immediately. Ironically, Rice had already in March 1995 told, in confidence, a few friends that he would resign as Head at the end of the summer. Rice announced his resignation as Head in late July of 1995, effective at the end of the 1995/96 academic year. Wayne Dyksen became Acting Head in May 1996. A search for a new Head was launched and Ahmed Sameh (from the University of Minnesota) became Head in January 1997.

The crises and change in Department Head had some belated positive effect on resources. The Dean allowed some of the empty positions to be filled and by the end of 1999, the size of the regular faculty had grown back up to 29. The Dean did not force Physics to transfer some space to Computer Sciences until after 2000. Firm limits on the number of entering freshmen in Computer

Sciences were finally put into effect in 2001. Only time will tell if the department prospers and regains some of its lost prestige.

7. Historical Data

The two 1994 papers by Rice and Rosen give considerable historical data for 1962–1990. These include the number of degrees awarded yearly, the full time faculty and professional staff. All the department's Ph.D. graduates are listed at http://www.cs.purdue.edu/phds/ along with their advisors and thesis titles.

ACKNOWLEDGEMENTS. Valuable discussion and input from Samuel Conte, Walter Gautschi, Felix Haas, Robert Lynch, and L. Duane Pyle are acknowledged.

References

1. M.J. Atallah, D.E. Comer, H.E. Dunsmore, G.N. Frederickson, and J.R. Rice. A Five Year Plan for Excellence. *CSD-TR 651*. Computer Sciences Department, Purdue University, 1986.
2. P.J. Denning, J.R. Rice, L. Snyder, and P. Young. A Plan for Excellence in Computer Sciences. *Internal Planning Memo*, Computer Sciences Department, Purdue University, July 15, 1981; Revised August 26, 1981.
3. P J. Denning, Eating Our Seed Corn, President's letter. *Comm. Assoc. Comp. Mach*., July, 1980.
4. J. Margarrel, As Students Flock to Computer Science Courses, Colleges Scramble to Find Professors, *The Chronicle of Higher Education*, Feb 9, 1981, p. 3
5. J.R. Rice and S. Rosen. History of the Computer Sciences Department of Purdue University. In *Studies in Computer Science: In Honor of Samuel D. Conte*, (R. DeMillo and J.R. Rice, eds.), Plenum Pub., pp. 45–72, 1994.
6. S. Rosen and J.R. Rice. The Origins of Computing and Computer Sciences at Purdue University. In *Studies in Computer Science: In Honor of Samuel D. Conte*, (R. DeMillo and J.R. Rice, eds.), Plenum Pub., pp. 31–44, 1994.

Appendix 1: Initial Computer Science Course Offerings - 1963/64

UNDERGRADUATE COURSES

200. Laboratory on Programming for Digital Computers.
Programming of digital computers in a problem-oriented language (FORTRAN). Problems solved will illustrate some of the elementary methods of numerical analysis. The course is intended to prepare the student to use digital computers in later courses in science and engineering.

400. Introduction to Programming for Digital Computers.
Stored program digital computers; programming in problem-oriented and machine-oriented languages; pseudo-operations, system macro-instructions, programmer-defined macro-instructions; solution of non-numerical problems.

GRADUATE COURSES

500. Computing and Programming Systems.
Computer organization as it affects programming. Magnetic tape systems. Error detecting and error correcting codes. Disc, drum, and other random access systems. Input-Output programs. Buffering, simultaneous operation, interrupt handling. Introduction to the design of compiling systems, Programming languages based on ALGOL. Backus Normal Form. Recursive procedures, dynamic storage allocation and other ALGOL features.

514. Numerical Analysis.
Finite difference calculus, finite difference equations; iterative methods for solving non-linear equations; differentiation and integration formulas; the solution of ordinary differential equations; the solution of linear systems; round-off error bounds.

515. Numerical Analysis of Linear Systems.
Computational aspects of linear algebra; linear equations and matrices, direct and iterative methods; eigenvalues and eigenvectors of matrices; error analysis.

520. Mathematical Programming.
Fundamental theorems from the theory of linear inequalities. Simplex Method and variants; gradient methods; special techniques for solving integer programming problems; survey of applications; survey of methods in current use on various computer systems; consideration is directed throughout to that class of algorithms created to solve mathematical programming problems which are appropriate for implementation on modern digital computers.

560. Information Storage and Retrieval
Computer-based information storage and retrieval systems. Selective dissemination of information; document retrieval. Indexing; file organization; search techniques. Automatic classification and abstracting. The structure of information systems.

580. Introduction to Data Processing
Problem solving and programming techniques and languages; students use computer-oriented and problem-oriented languages in solving problems common to the non-physical sciences.

(581. Introduction to Logic and Boolean Algebra.
See MA. 581)

582. Mathematical Theory of Finite Automata.
Structure theory of finite transition algebras and semi-groups, lattices of congruence relations on words; periodic sets of words; behavior of finite automata; transition graphs and regular canonical systems; Kleene's theory of regular expressions; Church's solvability-synthesis algorithm and its extension to wider classes of design requirements; decision procedure for sequential calculus and its significance to the algorithmic design of sequential machines.

583. Simulation and Information Processing.
Simulation and modeling. Monte Carlo techniques; use of special simulation languages to simulate actual systems; structuring and use of information systems; information storage and retrieval.

584. Recursive Functions I.
Post canonical systems, computability, decision problems; recursive functions, Turring machines, Markov algorithms, recursive algorithms.

(585. Mathematical Logic I. See MA 585)

600. Advanced Programming Systems I.
Design of assemblers, compilers, and interpretive systems. Libraries of subroutines, macro-instructions, and generators. Operating systems for debugging and job sequencing. Systems storage allocation. Programming languages and their translators. List processing languages. Generalized compiling systems for producing compilers.

601. Advanced Programming Systems II.
Continuation of CS 600.

614. Numerical Solution of Ordinary Differential Equations.
Numerical solution of initial-value problems by Runge-Kutta methods, general one-step methods, and multistep methods. Analysis of truncation error, discretization error, and rounding error. Stability of multistep methods. Numerical solution of boundary- and eigen-value problems by initial-value techniques and finite difference methods.

615. Numerical Solution of Partial Differential Equations.
The numerical solution of hyperbolic, parabolic elliptic equations by finite difference methods; iterative methods (Gauss-Seidel, over-relaxation, alternating direction) for solving elliptic equations; discretization and round-off errors; explicit and implicit methods for parabolic and hyperbolic systems; the method of characteristics; the concept of stability for initial value problems.

(616. Theory of Approximation. See MA 616)

(681. Artificial Intelligence. See EE 681)

(682. Theory of Automata. See EE 682)

684. Recursive Functions II.
Kleene hierarchies of non-recursive sets; advanced theory of recursivity, recursive un-solvability types; advanced topics on Turing machines.

Appendix 2: Undergraduate Computer Science Courses – 1967

LOWER DIVISION COURSES

210. Laboratory on Data Processing.
Programming of digital computers using problem-oriented languages (COBOL and FORTRAN). Problems will be selected from the areas of data processing and non-numerical applications. The course is intended to prepare the student to use computers in later courses in non-physical sciences.

220. Introduction to Algorithmic Processes.
Introduction to the intuitive notion of an algorithm; representation of algorithms in narrative form as flow charts and as computer programs; a general structure of computers; computer experience using a procedure-oriented language in programming algorithms such as those used in elementary numerical calculations, sorting, simulation of a random process and symbol manipulation; definition and use of functions, subroutines and iterative procedures; survey of a variety of significant uses of computers.

UPPER DIVISION COURSES

414. Introduction to Numerical Analysis
Iterative methods for solving non-linear equations; direct and iterative methods for solving linear systems; approximations of functions, derivatives and integrals; error analysis.

482. Discrete Computational Structures.
Finite and discrete algebraic structures relating to the theory of computers; semi-groups, groups, Boolean algebras. Directed and undirected graphs and their relation to these algebraic structures. Combinatorial problems inherent in computation.

484. Models for Algorithmic Processes.
Examples of various mathematical models for algorithmic processes; finite automata. Turing machines, Markov algorithms. Algebraic and graphical representation of these and other models.

(**485. Introduction to Mathematical Logic**. See MA 485)

UNDERGRADUATE AND GRADUATE

500. Computing and Programming Systems.
Computer Functional characteristics of the major units of a digital computer. Sequential and random access storage systems. Input-output channels, buffering, interrupt handling. Examination of the architecture of different computers. Computer organization as it affects programming. Number and symbol representation codes. Error detecting and correcting codes.

501. Programming Languages.
Syntax and semantics of several classes of programming languages (such as ALGOL, FORTRAN, SNOBOL, IPL-V, SLIP, LISP, COBOL, PL-1). Students are expected to write, debug, and run programs in several of the major languages discussed.

512. Numerical Methods for Engineers and Scientists.
A survey of the useful methods for computation. Solution of non-linear equations and systems of non-linear equations. Numerical methods for systems of linear equations. Approximate differentiation and integration. Numerical solution of ordinary differential equations. Introduction to partial differential equations and elementary approximation methods.

514. Numerical Analysis.
Iterative methods for solving nonlinear equations; linear difference equations, applications to solution of polynomial equations; differentiation and integration formulas; numerical solution of ordinary differential equations; round-off error bounds.

515. Numerical Analysis of Linear Systems.
Computational aspects of linear algebra; linear equations and matrices, direct and iterative methods; eigenvalues and eigenvectors of matrices; error analysis.

520. Mathematical Programming.
Fundamental theorems from the theory of linear inequalities. Simplex Method and variants; gradient methods; special techniques for solving integer

programming problems; survey of applications; survey of methods in current use on various computer systems; consideration is directed throughout to that class of algorithms created to solve mathematical programming problems which are appropriate for implementation on modern digital computers.

560. Information Storage and Retrieval.
Computer-based information storage and retrieval systems. Selective dissemination of information; document retrieval. Indexing; file organization; search techniques. Automatic classification of abstracting. The structure of information systems.

580. Design of Data Processing Systems.
Structuring of data processing systems and computer organization as it affects those systems. File organization; file maintenance routines; sorting, retrieval algorithms. Evaluation and analysis of batch processing, real time, and time shared systems and the related problems of feasibility and implementation. The total systems concept in the design of integrated information systems.

582. Mathematical Theory of Finite Automata.
Structure theory of finite transition algebras and semi-groups, lattices of congruence relations on words; periodic sets of words; behavior of finite automata; transition graphs and regular canonical systems; Kleene's theory of regular expressions; Church's solvability-synthesis algorithm and its extension to wider classes of design requirements; decision procedure for sequential calculus and its significance to the algorithmic design of sequential machines.

583. Simulation and Information Processing.
Simulation and modeling. Monte Carlo techniques; use of special simulation languages to simulate actual systems; structuring and use of information systems; information storage and retrieval.

583. Theory of Effective Computability.
The notion of an effective process; equivalence of various definitions. Turing machines, Markov algorithms, recursive functions, Post canonical systems; Church's Thesis, solvable and unsolvable decision problems.

(585. Mathematical Logic I. See MA 585)