CENTRO STUDI CALCOLATRICI ELETTRONICHE DEL C.N.R.

PRESSO L'UNIVERSITA' DI PISA

Nota Interna II. 61

INSEGNAMENTO UNIVERSITARIO NEL CAMPO DEI CALCOLATORI ELETTRONICI

A.GRASSELLI

Luglio 1967

1. Introduzione

A complemento di una precedente Nota Interna (II.43) si raccolgono qui alcuni documenti sull'insegnamento universitario nel campo dei Calcolatori Elettronici.

1. Corsi attualmente svolti nelle Università italiane.
2. Lista più recente di istituzioni americane che offrono corsi di laurea in Scienza dell'Informazione.
3. Alcuni commenti su corsi di laurea americani.
4. Sviluppi recenti nel campo dell'insegnamento di Scienza dell'Informazione in Gran Bretagna.
5. Corsi in Scienza dell'Informazione nelle Università britanniche.
6. Programmi di alcuni corsi svolti nelle Università britanniche.
7. Programma del corso in Scienza dell'Informazione all'Università di Grenoble.
8. Programma del Corso di Matematica Applicata al Politecnico di Grenoble.

1. Corsi attualmente svolti nelle Università italiane.

LAUREA IN CHIMICA

Indirizzo inorganico chimico-fisico

- Calcoli chimici e programmazione	- Univ. Parma
- Calcolo numerico e tecnica delle applicazioni meccaniche ed elettroniche	- Univ. Cagliari
- Teoria ed applicazione delle macchine calcolatrici.	- Univ. Cagliari

LAUREA IN ECONOMIA E COMMERCIO

- Statistica (biennale)	- Fondamentale sul piano nazionale.
- Econometria	- Univ.Catania, Genova, Messina, Catt."Sacro Cuore" Milano, Padova, Pavia, Pisa, Trieste, Lib. Urbino, Ist. Univ. Economia e Comm. Venezia.
- Organizzazione aziendale	- Univ. Bologna, Padova, Parma, Pavia, Perugia, Pisa, Torino, Inter. "Pro Deo" Roma, Ist. Univ. Economia e Comm. Venezia.
- Organizzazione e direzione aziendale	- Univ. Trieste.
- Principi e tecnica delle applicazioni meccanografiche ed elettroniche	- Univ. Trieste.
- Ricerca operativa	- Univ. Bologna, Cattolica "Sacro Cuore" Milano, Inter. "Pro Deo" Roma, Trieste, Ist. Univ. Econ. e Comm. Venezia.

- Teoria ed applicazione delle macchine calcolatrici.	- Ist. Univ. Econ. e Comm. Venezia.

LAUREA IN FISICA

- Calcolo numerici e grafici - Calcolo delle probabilità - Cibernetica e teoria dell'informazione	- Complementari sul piano nazionale.
- Analisi numerica	- Univ. Pavia, Roma, Trieste Lib. L'Aquila.
- Calcolatrici Elettroniche	- Univ. Palermo
- Calcolatori elettronici	- Univ. Roma
- Calcolo delle probabilità e statistica	- Univ. Torino
- Calcolo elettronico	- Univ. Catania, Lib. L'Aquila.
- Calcolo numerico	- Univ. Roma
- Calcolo numerico e programmazione	- Univ. Genova, Napoli, Parma.
- Cibernetica	- Univ. Catania, Napoli, Padova, Roma, Parma.
- Elaborazione e trasmissione delle informazioni	- Univ. Palermo
- Elettronica applicata alle macchine calcolatrici	- Univ. Napoli
- Logica delle calcolatrici digitali e teoria della programmazione	- Univ. Trieste
- Logica Matematica	- Univ. Bologna, Genova, Napoli, Parma, Pavia.
- Macchine calcolatrici	- Univ. Milano
- Ricerca operativa	- Univ. Firenze, Palermo

- Tecnica della programmazione	- Univ. Roma
- Teoria dell'informazione	- Univ. Genova, Napoli, Parma
- Teoria delle macchine calcolatrici	- Univ. Torino
- Teoria ed applicazioni delle macchine calcolatrici	- Univ. Bari, Bologna, Genova, Parma, Pavia.

LAUREA IN INGEGNERIA

- Controlli automatici	- Fondamentali sul piano nazionale
- Componenti elettronici	- Univ. Roma, Palermo, Padova
- Calcolo numerici, grafici, meccanici ed elettronici	- Univ. Pisa
- Analisi statistica	- Univ. Bologna
- Calcolo elettronico	- Univ. Bologna

LAUREA IN MATEMATICA

- Calcoli numerici e grafici - Calcolo delle probabilità - Logica matematica - Statistica matematica - Teoria dei numeri - Teoria ed applicazione delle macchine calcolatrici	- Complementari sul piano nazionale
- Algebra di Boole	- Univ. Padova

- Teoria della programmazione per le macchine calcolatrici - Univ. Padova
- Teoria delle macchine calcolatrici - Univ. Torino
- Teoria ed applicazione delle macchine - Univ. Trieste

US INSTITUTIONS OFFERING DEGREE PROGRAMS IN COMPUTER SCIENCE, DATA PROCESSING, AND RELATED PROGRAMS

The following list is taken from the recent publication *Computing in Higher Education—Expenditures, Sources of Funds, and Utilization for Research and Instruction, 1964-1965, with Projections for 1968-1969;* Southern Regional Education Board, Atlanta, Georgia. The publication is a report on the statistical survey carried out by the Computer Sciences Project of the SREB under the direction of Dr. John W. Hamblen and supported by the National Science Foundation. The data for the survey was reported during the period from September 1966 through March 1967.

The list below includes US institutions which reported degree programs in the computer sciences. It is reprinted in the *Communications* because of the many requests for such information which have been received in the ACM office.

The institutions are arranged alphabetically within state within name of the academic program (usually department name). The programs are categorized as being in one of four major programs, Computer Sciences, Information Science, Business Data Processing, and Scientific Data Processing, or in one of twelve options in other academic areas. The letters following the names of the institution signify Associate's (A), Bachelor's (B), Master's (M), and Doctoral (D) degrees.

Copies of the complete report can be obtained from SREB at a cost of $5.50 each.

Computer Sciences

Institution	1964-1965 A	1964-1965 B	1964-1965 M	1964-1965 D	Planned A	Planned B	Planned M	Planned D
Auburn University, Auburn, Alabama 36830						B		
University of Alabama, University, Ala. 35486						B		
University of Alaska, College Alaska 99735					A	B	M	
University of Arkansas, Fayetteville, Ark. 72701						B	M	
California St. Poly Col., San Luis Obispo, Cal. 93401						B		
Stanford University, Stanford, California 94305			M	D				
University of California at Berkeley, Berkeley, California 94720		B						
University of California, Davis, Davis, Cal. 95616						B		
University of California at Santa Barbara, Santa Barbara, California 93106							M	D
Yale University, New Haven Connecticut 06520		B	M	D				
George Washington University, Wash., D.C. 20006						B		
Brevard Engineering College Melbourne, Fla. 32924						B		
University of Florida, Gainesville, Florida 32601						B	M	
University of Miami, Coral Gables, Florida 33124						B	M	
Atlanta University Center, Atlanta, Georgia 30314						B	M	
Emory University, Atlanta, Georgia 30322								D
Georgia State College, Atlanta, Georgia 30303						B	M	
University of Georgia, Athens, Georgia 30601							M	
University of Hawaii, Honolulu, Hawaii 96822							M	
Bradley University, Peoria, Illinois 61606							M	
Northwestern University, Evanston, Illinois 60201							M	D
Southern Illinois Univ., Carbondale, Illinois 62901							M	D
University of Illinois, Urbana, Illinois 61822							M	D
Indiana State University, Terre Haute, Ind. 47809		B					M	
Purdue University, Lafayette, Indiana 47907		B	M	D				
University of Notre Dame, Notre Dame, Ind. 46556						*B		D
Iowa St. U. of Sci. and Tech., Ames, Iowa 50010						B	M	D
University of Iowa, Iowa City, Iowa 52240						B	M	D
Kansas St. U. Ag. and App. Sci., Manhattan, Kansas 66502						B	M	D
Morehead State College, Morehead, Kentucky 40351						B		
University of Kentucky, Lexington, Kentucky 40604						B		
University of Southwestern La., Lafayette, Louisiana 70501			M		A	B		
University of Maryland, College Park, Md. 20742							M	
Mass. Inst. of Technology, Cambridge, Mass. 02139		B	M	D				
Michigan State University, East Lansing, Michigan 48824						B	M	
Wayne State University, Detroit, Michigan 48202						B	M	D
Western Michigan University, Kalamazoo, Michigan 49001						B	M	
University of Minnesota, Minneapolis, Minn. 55455							M	D
Winona State Colege, Winona, Minnesota 55987						B		
Jackson State College, Jackson, Mississippi 39217						B		
Univ. of Southern Mississippi, Hattiesburg, Mississippi 39401					A			
Mississippi State University, St. Col., Miss. 39782							M	
Central Missouri State College, Warrensburg, Missouri 64093						B		
Univ. of Missouri at Columbia, Columbia, Missouri 65201						B	M	
Univ. of Missouri at Rolla, Rolla, Missouri 65401			M			B		D
Washington University, St. Louis, Missouri 63130			M	D				
University of Nebraska, Lincoln, Nebraska 68503						B	M	
University of New Hampshire, Durham, New Hampshire 03824							M	
Newark Col. of Engineering, Newark, N.J. 07102							M	
Princeton University, Princeton, New Jersey 08540		B	M	D				
Stevens Institute of Tech., Hoboken, N.J. 07030			M					D
New Mexico Inst. Mining & Tech., Socorro, New Mexico 87801						B		
New Mexico State University, University Park, New Mexico 88070							M	
Columbia Univ. All Campuses, N.Y., N.Y. 10027		B	M	D				
Cornell University, Ithaca, New York 14850						B	M	D
Pratt Institute, Brooklyn, New York 11205						B	M	
Rensselaer Poly Institute Troy, New York 12181							M	D
Suny at Albany, Albany, New York 12203					A		M	
Suny Downstate Med. Ctr., Brooklyn, N.Y. 11203						B		
Suny State Univ. of Buffalo, Buffalo, N.Y. 14214						B	M	D
N.C. State Univ. at Raleigh, Raleigh, N.C. 27607						B	M	
The Univ. of North Dakota, Grand Forks, North Dakota 58201						B		
Cuyahoga Community College, Cleve., Ohio 44115					A			
Ohio College of App. Science, Cinn., Ohio 45210					A			
Ohio State University, Columbus, Ohio 43210						B	M	D
University of Akron, Akron, Ohio 44304						B		
University of Dayton, Dayton, Ohio 45409		B					M	
Okla. State University, Stillwater, Oklahoma 74074						B	M	
Oregon State University, Corvallis, Oregon 97331			M	D		B		
University of Oregon, Eugene, Oregon 97403							M	
Carnegie Inst. Technology, Pittsburgh, Penn. 15212				D				
Lehigh University, Bethlehem, Pennsylvania 18015						B		
Pennsylvania State University, University Park, Pennsylvania 16802			M	D	A	B		
University of Pittsburgh, Pittsburgh, Pa. 15213							M	
University of Rhode Island, Kingston, R.I. 02881							M	D
Univ. of South Carolina, Columbia, S.C. 29208						B	M	
Winthrop College, Rock Hill, South Carolina 29730						B		
Vanderbilt University, Nashville, Tennessee 37203						B	M	D
Texas A & M University, College Sta. Texas 77840			M			B		
Rice University, Houston, Texas 77001							M	D
Sam Houston State College, Huntsville, Texas 77340						B	M	
Texas Christian University, Ft. Worth, Texas 76129					A	B	M	D
Texas Col. Arts Industries, Kingsville, Texas 78363						B		
Texas Technological College, Lubbock, Texas 79409							M	D
Univ. of Texas at El Paso, El Paso, Texas 78712						B		
University of Houston, Houston, Texas 77004						B	M	
Utah State University, Logan, Utah 84321		B						
University of Virginia, Charlottesville, Va. 22903			M	D				
Virginia Polytechnic Inst., Blacksburg, Va. 24061						B	M	
University of Washington, Seattle, Wash. 98105							M	D
W. Virginia Inst. of Technology, Montgomery, West Virginia 25136					A			
University of Wisconsin, Madison, Wisconsin 53706			M	D		B		
University of Wyoming, Laramie, Wyoming 82070						B		

Information Sciences

Institution	1964-1965 A	1964-1965 B	1964-1965 M	1964-1965 D	Planned A	Planned B	Planned M	Planned D
Univ. of Cal. at San Francisco, San Francisco, California 94122							M	D
Yale University, New Haven, Connecticut 06520							M	D
Georgia Institute of Tech., Atlanta, Georgia 30332			M					D
Illinois Inst. of Tech., Chicago, Illinois 60616							M	
University of Chicago, Chicago, Illinois 60637							M	D
Mass. Inst. of Technology, Cambridge, Mass. 02139		B	M	D				
Princeton University, Princeton, New Jersey 08540		B	M	D				
Cornell University, Ithaca, New York 14850			M					
Univ. of N.C. at Chapel Hill, Chapel Hill, North Carolina 27514			M					D
Ohio State University, Columbus, Ohio 43210							M	D
University of Akron, Akron, Ohio 44304						B		
University of Dayton, Dayton, Ohio 45409			M					
Western Reserve University, Cleveland, Ohio 44106			M	D				
Lehigh University, Bethlehem, Pennsylvania 18015			M					D
University of Pennsylvania, Phila., Penn. 19104			M	D				
Univ. of Texas at El Paso, El Paso, Texas 78712						B		
Washington State University, Pullman, Washington 99163			M					D

Business Data Processing

Institution	1964-1965 A	1964-1965 B	1964-1965 M	1964-1965 D	Planned A	Planned B	Planned M	Planned D
Gadsden Tech. State Jr. Col., Gadsden, Ala. 35904					A			
University of Arizona, Tucson, Arizona 85721			M	D				
Chaffey College, Alta Loma, California 91701					A			
College of San Mateo, San Mateo, California 94402	A							
East LosAngeles College, Los Angeles, Calif. 90022	A							
El Camino College, El Camino Cal., Calif. 90506	A							

(Continued from preceding page)

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
Fullerton Jr. College, Fullerton, Calif. 92632	A							
Palo Verde College, Blythe, California 92225					A			
Stanford University, Stanford, California 94305			M					
Cal. St. Poly Kello.Vohrs, Pomona, Calif. 91766						B		
Mesa County Jr. College, Grand Junction, Colorado 81501	A							
Southern Colorado State Col., Pueblo, Colo. 81005	A	B						
Thames Vly. St. Tech. Inst., Norwich, Conn. 06360	A							
Atlanta University Center, Atlanta, Georgia 30314						B		
Elgin Community College, Elgin, Illinois 60120					A			
Freeport Community College, Freeport, Ill. 61032					A			
Vocational Technical Institute, Carbondale, Illinois 62901	A							
Kansas City Kansas Junior Col., Kansas City, Kansas 66101	A							
Eastern Kentucky University, Richmond, Kentucky 40476					A			
Morehead State College, Morehead, Kentucky 40351					A			
Northeast Louisiana St. Col., Monroe, La. 71201						B		
Baltimore Junior College, Baltimore, Md. 21215					A			
Boston University, Boston, Massachusetts 02215						B		
Northeastern University, Boston, Mass. 02115	A							
Grand Rapids Junior College Gr., Grand Rapids, Michigan 49502	A							
Wayne State University, Detroit, Michigan 48202			M					
Itawamba Jr. College Voc. Tech., Tupelo, Mississipi 39942					A			
Mississippi State University, St. Col., Miss. 39782						B	M	D
Northwest Mississippi Jr. College, Senatobia, Mississippi 38668					A			
Central Missouri St. College, Warrensburg, Missouri 64093	A							
Missouri Southern College, Joplin, Missouri 64801					A			
Washington University, St. Louis, Missouri 63130	A							
Princeton University, Princeton, New Jersey 08540		B						
New Mexico State University, University Park, New Mexico 88070					A			
Auburn Community College, Auburn, N.Y. 13021					A			
Pace College, New York, New York 10038					A	B		
Cuyahoga Community College, Cleveland Ohio 44115	A							
Lorain Co. Cmty. College, Lorain, Ohio 44052	A							
University of Akron, Akron, Ohio 44304					A	B		
University of Toledo, Toledo, Ohio 43606					A			
Drexel Inst. of Technology, Phila., Penn. 19104		B						
Odessa College, Odessa, Texas 79760					A			
Wharton County Junior College, Wharton, Texas 77488	A							
Weber State College, Ogden, Utah 84403	A							
Richmond Prof. Institute, Richmond, Va. 23220					A			
Centralia College, Centralia, Washington 98531	A							
Columbia Basin College, Pasco, Washington 99301	A							

Scientific Data Processing

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
College of San Mateo, San Mateo, Calif. 94402					A			
Vocational Technical Institute, Carbondale, Illinois 62901					A			
Suny Upstate Medical Center, Syracuse, N.Y. 13210					A			
Lorain Co. Cmty. College, Lorain, Ohio 44052					A			

Option in Mathematics

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
University of Alabama, University, Ala. 35486						B	M	
University of Alaska, College, Alaska 99735		B	M					
University of Arizona, Tucson, Arizona 85721			M	D				
Cal. St. Poly Kello. Vohrs., Pomona, Cal. 91766						B		
Southern Colorado State Col., Pueblo, Colo. 81005		B						
Wesleyan University, Middletown, Conn. 06457			M	D				
Florida State University, Tallahasse, Fla. 32306		B	M					
University of Illinois, Urbana, Illinois 61822			M	D		B		
Boston University, Boston, Massachusetts 02215						B		
Washington University, St. Louis, Missouri 63130		B						

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
New York University, New York, New York 10003					A	B	M	D
Ohio State University, Columbus, Ohio 43210			M	D				
University of Oklahoma, Norman, Oklahoma 73069		B	M	D				
Brown University, Providence, Rhode Island 02912			M	D		B		
Clemson University, Clemson, South Carolina 29631		B					M	
University of Tennessee, Knoxville, Tennessee 37916						B	M	
East Texas State Univ., Commerce, Texas 75429		B						

Option in Electrical Engineering

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
Arizona State University, Tempe, Arizona 85281		B	M	D				
University of Arkansas, Fayetteville, Ark. 72701			M					
Univ. of California at Berkeley, Berkeley, California 94720		B	M	D				
Univ. of Southern California, Los Angeles, California 90007		B	M	D				
University of Connecticut, Storrs, Conn. 06268			M					
University of Illinois, Urbana, Illinois 61822			M	D				
Mass. Inst. of Technology, Cambridge, Mass. 02139		B	M	D				
New York University, New York, New York 10003						B	M	D
University of Oklahoma, Norman, Oklahoma 73069		B	M	D				
Virginia Military Institute, Lexington, Va. 24450						B		

Option in Applied Science

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
University of California Davis, Davis, Cal. 95616							M	D
Southern Illinois Univ.,Carbondale, Illinois 62901			M					

Option in Linguistics

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
Georgetown University, Washington, D.C. 20007			M	D				

Option in Systems and Communications Science

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
Mass. Inst. of Technology, Cambridge, Mass. 02139		B	M	D				
Carnegie Inst. Technology, Pittsburgh, Penn. 15212				D				

Option in Quantitative Analysis

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
University of Chicago, Chicago, Illinois 60637			M	D				
Southern Methodist University, Dallas, Texas 75222		B	M					

Option in Systems Engineering

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
University of Arizona, Tucson, Arizona 85721		B	M	D				
Old Dominion College, Norfolk, Virginia 23508						B		

Option in Machine Computers

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
George Washington University, Wash., D.C. 20006		B	M					D

Option in Systems Analysis

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
Miami University, Oxford, Ohio 45056		B						

Option in Administrative Science

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
Florida Atlantic University, Boca Raton, Fla. 33432		B						
Clemson University, Clemson, South Carolina 29631						B		

Option in Management Science

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
Univ. of Southern California, L.A., Calif. 90007			M	D				
American University, Washington, D.C. 20016			M					D
Pace College, New York, New York 10038							M	

Option in Information Systems

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
University of Minnesota, Minneapolis, Minn. 55455						B	M	
Lehigh University, Bethlehem, Penn. 19015							M	D
Temple University, Philadelphia, Pa. 19122							M	

Option in Industrial Engineering

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
Arizona State University, Tempe, Arizona 85281		B	M	D				

Option in Statistics

Institution	1964-1965				Planned			
	A	B	M	D	A	B	M	D
Emory University, Atlanta, Georgia 30322			M	D				
Princeton University, Princeton, New Jersey 08540						B	M	D

GRADUATE COMPUTER SCIENCE PROGRAMS AT AMERICAN UNIVERSITIES

F. S. Beckman

IBM Watson Research Center
Yorktown Heights, New York

ABSTRACT: Considerable activity is currently under way in the establishment of graduate computer science programs at many universities. Some of the problems encountered are described. A variety of different approaches to these problems in the areas of program content, organization, research and faculty selection are summarized for some leading universities.

Research Report
RC-1895
September 11, 1967

There is an enormous amount of activity under way at American colleges and universities which involves setting up or expanding programs in Computer Science. A recent (September 1966) informal list, available from the Association for Computing Machinery, includes approximately 63 schools that are now offering programs which might be considered Computer Science activities. However, based on the many omissions observed and the flurry of very recent activity, the number is surely closer to 100. The enormous demand for skilled workers in digital computing, coupled with the glamour and appeal that high speed computing seems to have for many young people, has led to considerable interest among college students in digital computing as a profession; the actions of many schools are, in part, responses to this.

Some notion of different perspectives that prevail can be obtained from the titles that are used to refer to these programs. Of those schools on the ACM list, some two-fifths identify their programs as Communications Science -- Computer Science -- Computer Technology -- Computer Theory -- Computers and Automata or Information (Processing) Science.

About one-sixth use such terms as Applied Mathematics -- Computer Oriented Mathematics -- Mathematical Methods and Computers -- or Numerical Analysis.

One-third emphasize commercial data processing and refer to their programs as Administration Sciences -- Business Administration -- Business, Electronic or Management Data Processing -- Industrial Engineering -- Quantitive Business Analysis -- or Techniques of Management. Most of these commercially oriented programs are conducted at the junior college or undergraduate level. I shall say very little concerning them.

Some one-tenth look at the "big picture" and emphasize the systems aspects of computers, using titles such as Systems Engineering -- Systems Science -- Systems Analysis or Operations Research and Systems Analysis.

I shall describe some of the principal problems that seem to be associated with Computer Science education and some of the different approaches that are used. Most of my comments will concern education at the graduate level, but some brief consideration of undergraduate courses and activity is necessary, if only to define that offering which is considered suitable for the graduate level.

A widely accepted definition of the information sciences is given in a report on the program at the University of Chicago. "The Information Sciences deal with the body of knowledge that relates to the structure, organization, transmission and transformation of information... This includes the investigation of information representation, as in the genetic code or in codes for efficient message transmission, and the study of information processing devices and techniques, such as computers and their programming systems."

Many people have pointed out that one significant and unifying characteristic of the subject matter appropriate to Computer Science is the involvement with the effective or algorithmic approach. The notion that "one can find" a number is more pertinent and meaningful than "there exists" a number. This is closely related to the underlying concern of Intuitionistic Mathematics.

Different Orientations

Information or Computer Science is, however, subject to a variety of different orientations and associated objectives. Surely, a pressing immediate need is to equip large numbers of professional workers to use effectively the tens of thousands of digital computers currently installed. However, many feel that the long range benefits of studying in depth some of the more theoretical aspects of digital computing justify the study of subjects that have little immediate bearing on today's problems.

Although our subject has been motivated largely by the existence and use of very real devices, with a consequent tendency to restrain an unduly abstract approach, there still appears to be a continuous range of activity, from the Pure to the Applied, in Computer Science. I shall illustrate such a spectrum of approaches by considering Automata Theory, a subject that is often considered to be well within the domain of Computer Science.

At the lowest (in the sense of least pure) level, Automata Theory is largely ignored. For example, there are no compelling reasons to teach parts of this subject in industrial education programs for programmers or computer systems engineers who will be concerned with the efficient use of computers and not with their design.

At a higher level, and I think this is illustrated by a typical approach in an Engineering School, Automata Theory may be identified occasionally with Switching Theory; and those parts of the subject dealing with the realization and optimization of automata are studied in detail.

At a third level, I would include the much more mathematically oriented work in Automata Theory and its cousin Mathematical Linguistics. Here, some of the fundamental concepts of modern algebra play a significant role; but, still, devices (Turing machines -- finite state automata -- one tape -- two tapes -- one way -- two way -- read only tapes -- push down stores -- linear bounded automata) are used to establish a structure that sometimes seems artificial and inelegant.

Finally, at a fourth and most pure level, I mention some very recent work, most of it not yet published, in applying ideas of categorical algebras to Automata Theory. While it is still much too early to evaluate this work, an attempt is being made to provide, for much of Automata Theory, a firmer mathematical structure which is not so obviously tied to the particular devices whose capabilities are being studied. I am thinking here of the kind of work to be represented in the Colloquium lectures of the American Mathematical Society which will be given by Professor Samuel Eilenberg of Columbia University at the Society's forthcoming summer meeting in Toronto. The meeting will be held in late August of this year and Professor Eilenberg will speak on "Universal Algebras and the Theory of Automata."

It appears to me that similarly differing approaches are taken to many computing subjects, and I have illustrated these only in the case of Automata Theory. These approaches reflect differing objectives which are not always clearly expressed. The kind of educational program developed at any school should certainly be closely related to its objectives. If the main goal is to prepare students to use computers more effectively, certain subjects -- those, for example, important to a program to train researchers in Computer Science -- would be de-emphasized. The attempt to train people to cope more successfully with complex systems development (which is, admittedly, very difficult: there is little convincing evidence that it has ever been done successfully) would be far more important than instruction in some of the more theoretical aspects of digital computing.

The following goals represent some of the different objectives in Computer Science education:

1. To train effective workers who will be included in the force of some hundreds of thousands of professional programmers and systems analysts who will be using digital computers in the very near future.

2. To train computer designers or "systems architects."

3. To train people who will have extensive knowledge of computing and those mathematical tools required to develop new ways of using computers. To equip them to recognize and to realize new and important applications.

4. To train a select few in the more theoretical aspects of Computer Science and other relevant subjects so that, by their research, they can further our understanding of effective procedures and the kinds of mental processes

that can be accomplished by them.

It may be appropriate to have University training to meet these goals, but it is probably not fitting to consider doctoral programs for all of them. Certainly, one of the expected results of this conference is to determine the desirability, goals, and content of a terminal professional course in Computer Science which does not extend through a doctoral program.

Subject Matter

Much work has been done on the identification and organization of the subject matter appropriate to Computer Science. The ACM Curriculum Committee on Computer Science has prepared an excellent and very well known first approximation to a Computer Science curriculum for undergraduate majors. (Communications of the ACM, September 1965) This report identifies fifteen courses that might comprise required and elective subjects for such a program. It also provides brief descriptions of the proposed courses and suggested reference materials. The titles of the courses are:

1. Introduction to Algorithmic Processes
2. Computer Organization and Programming
3. Numerical Calculus
4. Information Structures
5. Algorithmic Languages and Compilers
6. Logic Design and Switching Theory
7. Numerical Analysis
8. Computer and Programming Systems
9. Combinatorics and Graph Theory
10. Systems Simulation
11. Mathematical Optimization Techniques
12. Constructive Logic
13. Introduction to Automata Theory
14. Formal Languages
15. Heuristic Programming

Although I am not aware of any undergradua[illegible] hat now offers such a rich curriculum, many graduate programs i[illegible] of the material suggested by the ACM Committee. I understand tha[illegible]mittee will shortly issue a proposed curriculum for gr[illegible]uate programs [illegible]er Science (and I am sure that we shall hear a great deal about this in [illegible]shop discussions here).

While agreement on the subject matter of Compu[illegible]ence is possible, it is true that the field lacks the coherence of some [illegible]isciplines. Some people, disturbed by this and the highly specialized, pragma[illegible]proach often taken, prefer to think of it as a "science to be" -- confiden[illegible]at a more unified and scientific structure will appear.

In any event, in spite of this apparent amorphism and incoherence, I do not consider that the selection of subject matter constitutes the major difficulty in launching or maintaining a viable program. It is certainly necessary to be clear about the objectives of such a program (and I have specified several); but, having done this, I think that the selection of courses and course material, while not easy, is a quite solvable problem. There is, increasingly, a wealth of good published materials to support such courses.

Rather, I believe that the key problems are: 1) the setting of the objectives of the program (these, I think, are not always clear in existing programs) 2) the organization of the faculty entity that has the responsibility for the program and 3) the selection of the faculty. Here, it is of interest to note the following comment that appears on page 61 of "Goals of Engineering Education," October 1965, published by the American Society for Engineering Education. "Distinguished faculties are far more important to the advancement of engineering education than details of curricula or magnificence of facilities." Probably, a significant problem at the doctoral level is: 4) the identification of feasible research problems, the solutions to which require the kind of intellectual effort associated with the Ph.D. degree program -- when it is conducted at its best.

Many people believe that, apart from special graduate degree programs or undergraduate majors in Computer Science, it is desirable for colleges to provide every student with at least a basic understanding of computers and algorithmic processes. All informed citizens in an increasingly sophisticated society, where computers will play a universal and important role, should possess this understanding. Such "liberal arts" courses in "computer appreciation" are being experimented with and may become widely adopted. If so, there is here, too, the question of who in the faculty structure should have the responsibility for such courses.

Increasingly, apart from meeting the needs of the Computer Science majors, there is the problem of providing students in Physics, Chemistry, Biology, the Social Sciences, Psychology, Education, Linguistics, English Literature, Music, Urban Planning -- the list, of course, could extend through all departments at a University -- with the training and tools necessary to use computers effectively in their research programs. Based on my conversations with many Professors at Columbia University who are not in Mathematics and Engineering, I feel that, with regard to this problem, there is rather little interest in what seem to them to be the more esoteric aspects of Computer Science. Courses in programming and in the techniques of using computers for problem solving are considered very important, however.

Problems

I should like to mention briefly the more obvious problems involved in obtaining complete acceptance for graduate Computer Science education in some academic quarters.

For the sake of completeness, I mention the question of intellectual respectability. That this is a problem is clear from some of Professor A. Oettinger's remarks in the Communications of the ACM, December, 1966. There is probably no one here who requires answers to the doubts expressed by some members of the National Academy of Sciences about Computer Science as an independent discipline. These are cited in Oettinger's letter. 1) Why should there be a special graduate (let alone undergraduate) program in Computer Science any more than in the use of any tool such as in electron microscopy? 2) Almost all creative computer designers and software inventors have been trained either as Pure Mathematicians or as Experimental Physicists. 3) The training of faculty and students in computer usage can better be done by people in the various disciplines who have acquired computer experience rather than by a separate cadre of computer scientists. 4) It is not the business of universities to train computer center managers or systems experts. 5) Computer Science is not a coherent intellectual discipline but rather a heterogeneous collection of bits and pieces from other disciplines, including analysis and differential equations, linear algebra, mathematical logic, linguistics, information theory, decision and control, automatic control theory, circuit analysis, etc.

That impressions leading to these doubts about the professional status of Computer Science exist should not be too surprising. Such uncertanties concerning the intellectual substance of computing seem, in general, to emanate more from Mathematicians or pure scientists than from Engineers. I have heard this questioning of the suitability of graduate programs in Computer Science expressed by a man who is a distinguished and widely known pioneer in the application of computers to one of the pure sciences -- one who played an important role in the early use of digital devices for scientific calculation. I think the main reasons for this feeling, when it appears, are the enormous range of intellectual activity in computing, the many facets it exhibits, and unfamiliarity on the part of the observer with the full scope of this activity. Certainly, at its lowest levels there is much that is quite routine and pedestrian in digital computing. While acknowledging the importance of programming and the unusual intellectual qualities which are essential in an effective programmer, I can still imagine that we have all known examples of the programming virtuoso -- the very productive and brilliant programmer who produces successful code at ten times the rate of most of his fellows -- who has not, and seems not to need, a clear understanding of the basic theory of computability, of algorithmic processes, or of numerical analysis. His training may not have involved any significant amount of Mathematics or, if it did, there may be little correlation between his programming ability and the ability he displayed in his Mathematics courses. I in no way want to derogate the art of the programmer, but I can understand misgivings in some quarters about the academic respectability of such pursuits, especially when they are compared with study in Topological Spaces, Functional Analysis or Quantum Mechanics.

In the spectrum of computing activities, there is, of course, much work in Constructive Logic, Computability, and Automata Theory which involves a wealth of theorems, structure, and results that are far from obvious and which can be considered as advanced Mathematics. If a truly broad-minded approach is taken, the scope of activities relevant to computing may extend very far, indeed; the continuing study of effective processes may very possibly lead to a better understanding of human mental processes and consequent significant achievements in artificial intelligence. It may well be that a better understanding of the language of the brain, itself, will, as Von Neumann has speculated, lead to a much deeper understanding of Mathematics. This may be true especially with regard to the somewhat uneasy foundations of Mathematics.

As I have said, I have not detected any of the above doubts about the intellectual respectability of digital computing in the minds of Engineers or of many Applied Scientists. The obvious enormous importance of digital computing is, to them, sufficient justification for its inclusion in the graduate school program.

The levels of intellectual activity in computing are also quite evident in some publications related to the field. By way of example, and not in any particular order, I mention the following representative publications -- Datamation, The Data Processing Magazine, Proceedings of the IRE, The SIAM Review, The Journal of the ACM, The SIAM Journal, The Journal of Symbolic Logic and, The Transactions of the American Mathematical Society -- all of which contain, with varying frequency, articles and papers within the domain of Computer Science.

Everyone in the field is fully cognizant of the issues we have discussed here. Still, I should like to emphasize that this awareness of the great breadth of subject matter, and, therefore, of the different possible orientations, is crucial to the proper staffing and organization of any graduate educational activity. Equally important to the decisions about computing subjects to be included and the methods of presentation is a clear understanding of the objectives of the particular program. This situation is quite comparable to that in Mathematics, for example, where different programs, with different staffing and admission standards, should be followed at advanced levels, depending upon whether the objective is to accredit a University research scholar in Pure Mathematics, to yield teachers of elementary Mathematics, or to train Applied Mathematicians.

STAFFING:

Next, I shall mention a few of the problems of staffing Computer Science faculties. First, of course, there is the obvious shortage of experts in this very young field. I think this problem is most severe in those subjects of a more practical orientation. We have discussed industry's difficulty in finding people who can cope successfully with the increasingly ambitious and complex computer systems applications being developed. These applications often take several years to produce and there are not many people who have successfully navigated such experiences and acquired the distilled wisdom that, in part at least, might be communicated to students who have little practical experience. Also, there is such a jungle of activity in some fields -- such as programming systems -- that the talents and experience necessary to see coherent structure are not common.

There is a less obvious, but still significant, problem in setting up Computer Science staffs at some of our best Universities -- those which carefully guard their reputation for being intellectually distinguished. These Universities attempt to reserve the Professorial title for those who are widely recognized as distinguished leaders of their field. Now, it does seem to be possible to identify an elite in the corps of Mathematicians or Theoretical Physicists. At least, if a group of Mathematicians were asked, individually, to identify the ten best Mathematicians in the country, they would agree to a significant extent on some of the names, although there would naturally be considerable deviations of opinion. I do not believe, from a small sample poll I have conducted, that it is possible to so identify the unusually distinguished Computer Scientists. There are, as we well know, good people and mediocre people, but I do not believe that, at the upper end of the scale, the best people are quite as far above the mean as their counterparts in Mathematics or Physics. While Von Neumann, and perhaps Turing, would appear on many lists, if they were alive, I have found surprisingly little agreement on the selection of the very best people currently active in computing. This kind of identification seems to be particularly difficult in such obviously practical subjects as the design and development of computer and programming systems. Here, I think the situation is similar to that in Engineering. In subjects like Automata Theory or Mathematical Linguistics, there is, however, less of a problem.

ON SUITABLE DOCTORAL THESIS PROJECTS:

One misgiving that I have often heard expressed concerns the selection of suitable Ph.D. research topics in Computer Science. Apart from the more mathematical parts of the subject, what might constitute, for example, a good Ph.D. thesis in Programming Systems? Would a project that requires a great deal of work but comparatively little in the way of new insights or creative ideas be suitable? Again, it is usually the Mathematician or Theoretical Physicist who feels most unsure of the acceptability of this kind of project for doctoral work. On the other hand, the Professor of Experimental Physics, of Applied Mathematics or of Engineering, who has seen many doctoral theses accepted for which the strongest raison d'être has been the substantial labors of the student leading to some small contribution, is usually not so doubtful of the acceptability of such thesis projects. I have, for example, heard a doubt expressed as to whether such an outstanding piece of work in computing as the specification of the ALGOL language would constitute an acceptable project if it were done, for the first time, by a doctoral student. Standards of acceptability are not easy to define; but an understanding by the Pure Mathematician or Theoretical Physicist, who has become a Computer Scientist, of the criteria that are prevalent in many of the sciences and in Engineering may be desirable.

ORGANIZATION:

One of the most crucial questions in starting a new program concerns the faculty organization for a Computer Science program. There have been many different approaches taken in Universities around the country. Should there be a separate Computer Science Department, as I believe is preferred at this time by a very large number of those who think of themselves as Computer Scientists? If there is such a department, should it be in the Engineering School or in the Graduate School of "Arts and Sciences"? In some schools, such as Purdue, the department may exist as a separate entity but within a Division (or School, or Faculty, or Institute, or Center) of Mathematical Sciences. There are also Institutes of Mathematical Sciences at Stanford, New York University, and Carnegie Institute. At a very new University like the University of Waterloo in Western Ontario, Canada, which first admitted students in 1957, there is a substantial Faculty of Mathematics divided into five departments including a Department of Applied Analysis and Computer Science, a Department of Applied Mathematics, a Department of Combinatorics and Optimization, a Department of Pure Mathematics and a Department of Statistics. Such an approach would probably be very difficult to implement at an older school which has a certain amount of rigidity in its existing departmental structure. At Columbia University, for example, the department is a very strong structure extending through the Undergraduate College, Graduate School and School of General Studies. In addition to these programs, the Mathematics Department sponsors various service courses which are open to students in the professional schools of the University. The setting up of new departmental structures, when the above situation prevails, and when there is some overlapping of interests with Mathematics and Engineering, would seem to present formidable problems.

The existence of a separate Computer Science Department is felt by some to be of special importance with respect to the recruiting of new staff members. In particular, new Ph.D.'s whose degrees are in Computer Science may attach considerable importance to the University's acceptance of Computer Science as an independent discipline as evidenced by giving it full departmental status.

Another argument that is heard in support of a separate Department of Computer Science is that a faculty can be assembled which will include specialists in many of the areas that are significant in computing, thus giving the student the opportunity to get a varied exposure, and encouraging that stimulating interdisciplinary exchange that is often so fruitful in computing and its applications. However, it is not at all clear that Computer Science Departments of the "second generation" -- those which will have staff members who have their degrees in Computer Science and not, as today, in Mathematics, Physics, Engineering, Linguistics etc. -- will provide this cross fertilization as effectively.

Should, perhaps, Computer Science be an interdepartmental activity under the control of a central committee? Again, should this committee be within the Engineering School or the Graduate School of Arts and Sciences? The feasibility of such an approach seems to vary with the school. At the University of Chicago, a committee in Information Science exists; but I understand that it has almost departmental status and that many of its members have joint appointments with such departments as Applied Mathematics, Physics and Physiology.

Should Computer Science be offered as an option under the Electrical Engineering Department as it is at the University of California in Berkeley or at Princeton? Or, should it be an option under the Mathematics Department? This latter approach does not now appear to occur with any great frequency, although some smaller schools like Stevens Institute adopt it. At Columbia, the Mathematics Department, which is quite pure, has no interest in such an arrangement; but it may be more attractive in an Engineering College, or at a University like Brown where there is a separate Department of Applied Mathematics, or at Harvard where, although there is no independent Department of Applied Mathematics, there are Professorial appointments in Applied Mathematics.

It is interesting to note that at the University of California in Berkeley a separate mathematically oriented Department of Computer Science is, apparently, being established within the Graduate School of Arts and Sciences. That University will thus have two independent graduate programs in Computer Science.

Or should there be, as some of us are beginning to believe, a Computer Science Institute (or Center or School) which includes the Computer Center and is authorized to offer courses and award degrees? At Columbia some institutes have been established in certain areas of study which have strong interdisciplinary flavor. An example is the Russian Institute which is associated with Columbia's School of International Affairs. Students pursue diversified courses of study, and their program for an advanced degree under the Graduate Faculties may include work in the Institute.

Such an institute could include a program of continuing educational activity -- seminars, lectures, and workshops -- that would be of benefit to the great community of computer users at a large University. The scope of such activities would considerably exceed that which is normally associated with the Computer Center, alone. It would, of course, also be desirable to have some research activities going on at such a Center. The approach at the University of Maryland seems to be related to this. Here, there is a Computer Science Center which acts as an interdisciplinary department. It is not affiliated with any school or college of the University, but provides service to them all. Also, the Georgia Institute of Technology, with its School of Information Science, may be moving in this direction.

I doubt that, at this time, there is any single, correct arrangement which is applicable to all schools. To a large extent, this is true because of the artificial restraints imposed by local organizational and political restrictions. In any event, it appears to me that there is no intellectual discipline represented at a University that has quite the interdisciplinary character of Computer Science. I believe that any organizational structure must recognize this and encourage the continuing development of new applications in now unrecognized areas.

Almost all of the computer users with whom I have spoken at Columbia, who represent many departments including Physics, English Literature, Economics, Civil Engineering, Sociology and others, were universally enthusiastic about extending activities in computer education. However, as I have said earlier, they were much more concerned with the additional support that this would provide to those of their graduate students who use computers in their research projects than they were with any ambitious program in Computer Science, *per se*. In fact, a few people expressed some concern that the preoccupation of Computer Scientists with the more abstract parts of their subject might detract from those activities that would facilitate the current use of computers and the development of new applications.

Some of the reluctance on the part of some Mathematicians to accept Computer Science as an independent discipline, on a par with Mathematics or Physics, appears to be closely related to the friction that is sometimes evident between Pure and Applied Mathematics. Here, it is relevant to note that Mathematical Statistics has to a large extent received recognition as a discipline worthy of separate departmental status, although some of the most untarnished Mathematicians may still regard it as an activity akin to plumbing.

I should like to summarize briefly the organizational approaches being taken at some major Universities. I shall consider them in an order that is determined, for the most part, but not entirely, by a composite ranking, using the relative evaluations obtained in the recent (1966) study of graduate education by the American Council of Education. Since, at this time, it would be quite premature to judge the effectiveness of existing Computer Science programs, this composite ranking is based upon the assessments, in this study, of the Faculties of Electrical Engineering, Mathematics and Physics.

First is the University of California at Berkeley. Here, as mentioned earlier, the principal activity now under way in Computer Science involves programs conducted by the Electrical Engineering Department. This is a very strong department with, as of the time of my information, some 93 faculty members of whom about 30 seem to have a very strong interest in Computing Science. Students with a Bachelor's degree in Engineering can work toward an M.S. or Ph.D. in Engineering, while students with a degree in Mathematics or the sciences can work toward the corresponding Master's or Doctor's degree in Engineering Science. There also seem to be two different suggested curricula. One emphasizes

Machine Organization and Programming Systems while the other, in the "Theoretical Computer Sciences", concentrates on switching and automata theory, formal languages and the applications of coding techniques.

Professor Zadeh, Chairman of the Department of Electrical Engineering at Berkeley, presented arguments for including Computer Science programs within Electrical Engineering in a paper -- "Electrical Engineering at the Crossroads" -- which he presented to the IEEE in 1965. It is of great interest to note that he now feels that, in the long run, only a structure "like an institute" will have stability. (For further opinions of Professor Zadeh on these questions, see his paper in these conference proceedings.) As mentioned earlier, another Department of Computer Science is being established in the Graduate School of Arts and Sciences. It is likely that this new activity will draw most of its students from Mathematics.

At Harvard, judging from that institution's response to my requests for information, the approach to Computer Science is very casual. At the time of my inquiry, there was no glossy brochure available describing the computer-related educational activities at Harvard. There is, rather, an informal list of courses selected from the areas of Natural Sciences, Applied Mathematics, Engineering, Engineering Sciences, Linguistics, Business Administration and the Graduate School of Design and Education. This list is set forth as comprising the educational activity in Computer Science. A student may work toward a degree in Applied Mathematics, concentrating on courses in this area.

At M.I.T., Computer Science is one of some fourteen fields of specialization mentioned within Electrical Engineering. This is an enormous department with 113 members having the rank of Assistant Professor or above. At least thirteen courses relevant to Computer Science are offered within Electrical Engineering, alone. The Mathematics Department at M.I.T. is rather pure, and only two courses in Logic and Proof Theory appear to be relevant to Computer Science.

In the School of Humanities and Sciences at Stanford, there is a new Computer Science Department which offers Master's and Doctor's degrees. Some aspects of this program have been fully described by Professor George Forsythe in his paper, "A University's Educational Program in Computer Science" (Communications of the ACM, January, 1967). A recent Department bulletin lists 11 members of the Department with the rank of Assistant Professor or above. Professor Forsythe's paper also includes some interesting comments concerning the nature and objectives of Computer Science education.

At the California Institute of Technology, a new Department of Computer Science is, I understand, in the process of being formed.

As I mentioned earlier there is, at the University of Chicago, a "Committee on Information Sciences" which has almost departmental status. There are nine staff members with the rank of Assistant Professor or higher. Many of these seem to have joint appointments. In addition to appointment to the Committee, four of them are listed as Professors of Applied Mathematics, three as Professors of Information Sciences, one as a Professor of Physics and one as a Professor of Physiology. Some 19 courses are listed under Information Sciences, and a student can work toward either the S.M. or Ph.D. degree. Although there are some joint appointments, the committee seems to be independent not only of the Computer Center (this separation between the educational and service functions is prevalent at most schools) but also, somewhat surprisingly, of the Institute for Computer Research at Chicago.

Princeton University has a special program under its Department of Electrical Engineering and lists some 11 undergraduate and graduate courses in Computer Science which have Electrical Engineering titles.

At the University of Illinois, there is a new Department of Computer Science and, as of November 1966, it is authorized to award both the M.S. and Ph.D. degrees in Computer Science. An undergraduate degree in Mathematics and Computer Science is an option under the Mathematics Department. Prior to the formation of the Department, the Digital Computer Laboratory had obtained department status in 1959, but no degree program in Computer Science was available. The name of the laboratory was changed to the Department of Computer Science in 1965. The emphasis at Illinois seems to be on Computer Organization and Design; and there is, of course, a long and distinguished history of activity in this area, extending from the ORDVAC machine built in 1949 and including the ILLIAC's I, II, III, IV. There appear to be nineteen staff members, and, although all but one are full time members of the Department, most of them, for historical reasons and personal preference, carry titles associated with other disciplines such as Professors of Applied Mathematics, Electrical Engineering, Mathematics, Physics. Some twenty-three courses for the Computer Science major are listed. There are also research programs for students pursuing graduate degrees in other departments.

At Columbia University there is a Computer Science program under the Department of Electrical Engineering. Some thirteen courses in Electrical Engineering are relevant to Computer Science. Candidates for the Master's, Professional and Doctoral degrees can concentrate on courses in this area. There is an attempt, also, to keep admission requirements for doctoral candidacy flexible enough to accommodate students with undergraduate degrees in the sciences or Mathematics, as well as in Engineering. Many of the courses are given by visiting staff members, and a very small number of the members of the permanent staff have their principal interest in Computer Science. It is also possible for a student to work toward a degree in Computer Science under the supervision of an interdepartmental committee on <u>Mathematical Methods in Engineering and Operations Research</u> within the School of Engineering and Applied Science. There is a similar committee on Applied Mathematics within the Graduate Faculties which also, on occasion, oversees the work of a student who is doing some research relevant to Computer Science. As usual, the Computer Center activity is largely separate from the educational programs.

In the School of Graduate Studies at the University of Michigan, Computer Science is subsumed within an interdepartmental Communication Sciences program. A broad offering of courses in Formal, Artificial and Natural Systems is available, and the student qualifying for the Master's or Doctor's degree will generally take courses not only in the Communication Sciences Department but also in Electrical Engineering, Mathematics and Psychology.

A Computer Sciences Department was created at the University of Wisconsin in 1964. It currently has a staff of 29, many of whom hold appointments with Electrical Engineering, Mathematics, Linguistics, Theoretical Chemistry, Internal Medicine, English, the Mathematics Research Center, Computer Center, and the Library. Programs leading to the B.S., M.S., and Ph.D. degrees are available. A large number of courses are offered, most of which are divided into three areas. Some twelve courses are listed in the area of <u>Numerical Analysis and Mathematical Programming</u>. Under <u>Systems Programming and the Theory of Computation</u>, eleven courses appear, and thirteen courses are listed under <u>Models of Intelligence and Natural Language Processing</u>. Also, a large number of peripheral courses offered by other departments are mentioned in the description of the Department's program.

At Cornell University, an intercollege Department of Computer Science was established in 1965. Interestingly, the head of the Department reports to both the Dean of the College of Arts and Sciences <u>and</u> to the Dean of the College of Engineering. The main emphasis at Cornell is on graduate work and the student can work toward the M.S. or the Ph.D. degree. Some four elementary courses are listed as being primarily for undergraduates, with approximately thirteen courses intended for graduate students. There are eight members of the Department with the rank of Assistant Professor or above, and one visiting professor. Some six members of other departments also teach Computer Science courses or are engaged in relevant research activities.

At Yale University there are two Computer Science programs. One, within the Department of Engineering and Applied Science, enables the student to work toward the Ph.D. in the area of "Computers, Communications, and Control". Some eleven staff members of the Department are listed as having their principal interest in this area. There is, in addition, an interdepartmental committee, with representatives from the Departments of Engineering, Philosophy and Linguistics, which oversees the work of some students working in Computer Science. Some staff members hope for separate departmental status for Computer Science in the future.

The University of Pennsylvania has a graduate program in the Computer and Information Sciences within the Moore School of Electrical Engineering. The participating faculty includes some fifteen members from Electrical Engineering, two each from Linguistics, Mathematics, Philosophy and Economics, and five members from the Psychology Department. The program is supervised by a Graduate Group Committee. The interests and research activities of the faculty are quite broad and cover both hardware and software design and the more mathematical parts of computer theory. A student can work toward the M.S. or Ph.D. degree.

Purdue University, as mentioned earlier, contains a Division of Mathematical Sciences which consists of three departments: Mathematics, Statistics, and Computer Sciences. There are some eight members of the Department of Computer Sciences with the rank of Assistant Professor or higher, and there is one visiting staff member. Master's and Doctoral programs are available in three areas of specialization: Numerical Analysis, Programming and Systems, Logic and Automata. Approximately twenty graduate courses are listed.

A Department of Computer Science was formed in 1965 at Carnegie Institute. It offers only the Ph.D. degree. Some nine staff members with the rank of Assistant Professor or above are listed, together with two visiting staff members. Many of these have joint appointments with other departments, including Mathematics, Electrical Engineering, Architecture and Industrial Administration. Undergraduate programs related to Computer Science are offered by the Department of Mathematics (Mathematics of Computation Option), by the Department of Electrical Engineering (Applied Computer Sciences Option), and by the Administration and Management Science Department (Computer Science Option). Nine graduate courses are listed as being offered by the Department of Computer Science. It is of interest to note that the usual foreign language requirements for Ph.D. candidacy have been replaced by the requirement that students show mastery of three programming languages.

At the time I gathered my information, a Computer Science Committee at the University of Washington in Seattle intended to offer, beginning this fall, a formal degree program leading to the M.S. and Ph.D. degrees. Approximately 27 courses relevant to Computer Science, of which about twelve have not yet been offered, are listed in the tentative program description. Most of these courses have Computer Science listings, although there does not appear to be a Department of Computer Science. About six of the listed courses -- in Algebra, Logic, Operations Research, and Numerical Analysis -- are offered by the Mathematics Department. The fifteen Committee members all seem to be members of other departments. The departments represented are: Mathematics, Electrical Engineering, Civil Engineering, Psychology, Physiology and Biophysics, Chemistry, Finance and Statistics, and Physics.

As I mentioned earlier, there is a Computer Science Center at the University of Maryland. It was established in 1962 as an interdisciplinary department, not affiliated with any school or college of the University. The Director of the Center reports to the Vice-President for Academic Affairs. The Computing Center is an integral part of this activity and the head of the Computing Center is an Associate Director of the Computer Science Center. One cited advantage of this arrangement is that it eliminates competition between the Computing Center and a separate department for research funds and staff. The interdisciplinary graduate educational program has developed gradually. Until now, there has not been a formal degree program in Computer Science, but, starting this fall, a Ph.D. degree program will be available. During the 1966-67 school year, some seven courses with Computer Science listings were shown. Faculty members from other departments serve the Center, on a part time basis, as Computer Research Consultants. These consultants supplement the Center's full time professorial faculty, of about a half dozen, who work in the areas of research and education.

At the University of North Carolina at Chapel Hill, a Department of Information Science was established in the College of Arts and Sciences in 1964 to offer graduate level instruction and pursue research in Information Science. An M.S. program is now offered, and students seeking the Ph.D. are being accepted. The Department does not offer any undergraduate major, but some courses are open to undergraduates. It is expected that some joint research projects involving other departments will be implemented in the future. Currently, the Department has two professors with joint appointments, one in English and Information Science and another in Sociology and Information Science. Some nine faculty and research staff members are listed. Of these, two have full time professorial appointments with the Department. The others have adjunct or visiting relationships, or a joint appointment with another Department or with the Computation Center. Some twenty-six courses are listed for graduate students, with about half of these being cross-listed with Mathematics, Linguistics, Philosophy, or Statistics. A few rather unusual courses bear such titles as Computational Stylistics ("the use of the computer in the specification and discrimination of stylistic patterns in" the aesthetics) and a Seminar in Teaching and Professional Practice. The University shares the use of the Triangle Universities Computation Center with Duke University and North Carolina State University and also has a smaller computing facility for its own use.

My very brief description of some of the salient characteristics of Computer Science programs at these nineteen schools is intended to be suggestive and not inclusive. Because of the lack of time, I have omitted references to many interesting programs, but my descriptions do provide an indication of the great diversity of organizational approaches and of the importance of inter-disciplinary relations in Computer Science educational and research activities. It has been estimated that within three years there will be over 200 colleges and universities in this country offering educational programs in Computer Science.

In concluding, I should like to quote briefly from the publication of October, 1965, "Goals of Engineering Education," prepared by the American Society for Engineering Education. This is the report of a study concerned largely with graduate programs in Engineering. After pointing out that ten to twenty years will be needed to develop a well established graduate program, and after reviewing the development of many of the more successful programs this statement is made:

"Perhaps the most important factor was the importance of individual initiative on the part of one or more highly motivated key persons -- and perhaps the primary lesson to be learned from this study is the importance of outstanding faculty people who have clear goals."

REVIEW OF RECENT DEVELOPMENTS

Five years ago it would have been difficult to point to any widespread systematic education of people working in the computer field. Today one can see a pattern of such education arising, with some awareness both of the types and numbers of personnel which are required and of what they should be taught. That this should have taken place in a relatively short period is witness to the urgency of the need and to the response which this has brought forth from our long established educational bodies.

It is appropriate that this publication should follow closely on a report on 'Computer education' prepared by an Interdepartmental Working Group.[1] This report examines in much greater detail than is possible here the present estimates of demand, the pool of available ability, the existing educational provision, and the steps needed to meet the demand. By way of introduction to these problems we shall try to summarize the basis and conclusions of the report.

In its quantitative aspects the report relies substantially on the estimate that the number of computers of varying sizes in operation or on order in the United Kingdom by 1970 will be over 3,000, to which may be added about 1,000 more in the planning stage and needing some staff. These numbers, perhaps intentionally somewhat conservative, refer mainly to computers in industry and commerce. In estimating staff numbers a very broad classification is used which includes

advanced programmers (designers of computer software systems)

systems designers and systems analysts (concerned with applying computers within a particular organization)

programmers (concerned with writing and checking detailed programs)

operators (for computer operation and data preparation)
maintenance staff.

This is a fairly satisfactory classification in this context. It is assumed that advanced programmers and systems designers are of honours degree standard, that most systems analysts are of graduate quality or close to it, and that the other classes mainly fall within technical grades. The requirements within these classes by 1970 are then estimated as follows:

	Numbers in 1964[2]	*Additions needed by* 1970
Advanced programmers	(Not specified)	200
Systems designers	3,600	500
Systems analysts	3,600	11,000
Programmers	4,800	19,200
Operators	3,000	16,000
Maintenance staff	2,500–3,000 (in 1965)	(Not specified)

Several points must be stressed. First these figures refer to the needs of industry and commerce, and do not allow for those of academic research and teaching establishments. Secondly, the view is expressed that the preliminary training of programmers is not a serious problem. The report states that 4,000 new programmers were trained by computer manufacturers in 1964, 6,000 more were expected in 1966, and the manufacturers could meet the added needs of 1970. Similar optimism was shown for operators and maintenance staff. However the more highly qualified staff, especially systems designers and analysts, presented a probable area of shortage.

An immediate impression given by these estimates is that the number of advanced programmers proposed to meet future demands for computer software systems is too low, even allowing for the fact that non-industrial needs have been excluded. The needs of universities, especially the regional computer centres, and many research establishments for such staff is considerable. Much effective software has been and will continue to be developed in universities. Moreover the potential demand for more complex operating systems associated with multi-processor systems and computer networks (as contemplated by the Post Office, banks and other large organizations) is so great that we should expect a much larger addition to the strength of software specialists to be required by 1970.

The report includes 17 paragraphs of recommendations and we can refer briefly to the most relevant:

Universities: Greater allocation of computer facilities for teaching purposes. Joint planning of high level courses with computer manufacturers. Reorientation of some courses to meet the need for systems designers. Increased numbers of grants to fill existing courses, especially to non-science graduates.

R. A. BUCKINGHAM

IN COMPUTER EDUCATION

Colleges of Further Education: Review of existing syllabuses. Continued support for and extension of courses such as City and Guilds 319 and 320. Supplementary courses for specialists in other fields. More short courses for existing teachers.

Schools: Increasing provision of desk calculators and other mathematical equipment. Further limited experiment with new mathematical syllabuses, and use of travelling computers. Organized access to computers, especially in colleges of further education, where time should be allocated to schools.

Throughout there is emphasis on the high priority to be given to courses for systems analysts, and the roles of universities, colleges and the *National Computing Centre* in this respect.

In the remainder of this article we shall look at some of the progress which has been made in recent years, especially in universities and colleges, and the prospects of meeting the needs of 1970.

Computer Science

a) Postgraduate Courses

In the universities specialized courses in methods of computation have had quite a long history and it would certainly be possible to trace their development through the 1920s and 1930s. The war years gave considerable impetus to the investigation of numerical techniques and it was natural that these should become the subject of postgraduate diploma courses in a number of universities soon after the war. Thus in the decade from 1950 to 1959 the emphasis was very definitely on the side of numerical analysis, which at that time was undergoing a good deal of development. However, with the arrival of computers in a few universities an introduction to programming soon crept into the courses and this began a process of change which has led to their becoming more and more computer-oriented.

Since 1964 the number of such courses has increased and the majority now aim to provide an M.Sc. degree rather than diploma, see *Survey of Computer Science Courses* in this review. A glance through the more detailed information which is presented later will show the extent of the shift away from numerical techniques and numerical analysis and towards topics such as theory of computation, logic design of computers, symbol manipulation, theory of computer languages, compilers, data structures and information processing. Typical examples may be seen in the syllabuses for Cambridge, Glasgow and London, and the trend is particularly evident in the more specialized courses at Edinburgh and St Andrews. At the same time it should be added that the opportunities for students to study numerical analysis have also improved, as this is now an accepted option in many Diploma and M.Sc. courses in mathematics.

It is important to note the numbers of students involved. Although the numbers given in this *Bulletin* are only approximate, and the list of courses may not be complete, it appears that between 200 and 250 students are registered for Diploma and M.Sc. courses in 1966–67, about one-third of these in London. There are other courses not included in the *Bulletin* survey. Many are in an early stage, so it seems reasonable to hope that by 1970 the number of postgraduate awards will exceed 500 per annum, provided financial support for the students is made available.

No comparable survey has yet been made of postgraduate students following a programme of research in computer science. The number is undoubtedly increasing and becoming significant and it must be hoped that details will be available for the next AER. Information is also incomplete on postgraduate courses suitable for the training of systems analysts and designers. These are still few, but mention should be made of the Diploma course on data processing in business administration at the *University of Newcastle*, and in London, to an M.Sc. course in systems analysis at *Imperial College*, and a future M.Sc. course in systems analysis at the *London School of Economics.* The emphasis given in the report on Computer Education to the need for reorientation of existing courses and for new courses for system designers suggests that a development is imminent in this field similar to that in computer science.

b) First Degree Courses

It is to be expected that in the older universities postgraduate courses in computer science and related subjects would develop in advance of first degree courses. Experiments in new fields of study, especially where the subject matter is rapidly evolving and the boundaries are ill-defined, naturally occur at the higher level. Moreover there has been considerable debate as to whether computer science represented an acceptable new discipline, or was better regarded as a supplementary study to be pursued after a first degree course in one of the well-established subjects.

This argument is now being resolved quite rapidly in favour of first degree courses which are designed either to be predominantly concerned with the theory and use of computers or to contain a substantial proportion of computer science in conjunction with other subjects. For the first of these alternatives, the lead has mainly been taken by six colleges of technology which have obtained approval of the *Council for National Academic Awards* to offer a B.Sc. degree in computer science. These are four-year sandwich courses which include industrial experience and have usually been developed in association with the computer industry; honours degrees will be awarded, although at several colleges (and perhaps in time at all of them) degrees at ordinary level will also be available. Among the universities Manchester has also been a pioneer in developing computer science as a full degree subject.

At a number of universities a joint honours or general honours degree which includes computer science is the favoured alternative. The overall present position seems to be as follows:

B.Sc in Computer Science	*No. of years*	*First graduates expected*
University of Manchester	3 (hons)	1968
University of Newcastle	3 (hons)	1969
Heriot-Watt University	3 (hons)	? 1970
University of St Andrews	3 (ord), 4 (hons)	? 1970
Brighton C of T	4 (hons)	1968
Hatfield C of T	4 (hons and ord)	1969
Staffordshire C of T	4 (hons and ord)	1969
Wolverhampton C of T	(4 (hons)	1969
Constantine C of T	4 (hons and ord)	1970
Leicester Reg. C of T	4 (hons)	1970

B.Sc (Joint or General Hons., 3 *years)*

University of Leeds	Joint Hons. with physics or mathematics
University of Newcastle	Gen. Hons. (½ computer science)
University of Reading	Gen. Hons. (⅓ computer science)

B.Sc (with variable proportion of computer science, 3 years)

University of Glasgow
University of London
(particularly at Birkbeck C, LSE, Queen Mary C)

B.A in Mathematics (3 years)

University of Lancaster	About ½ computer science
University of Leeds	Final year specialization in computer science
University of Liverpool	Final year specialization in computer science.

Clearly this list, especially the second half, is likely to lengthen as time goes on.

Again we should look at numbers of students. In 1966–67, at colleges of technology, there are 213 first year students enrolled in six courses, compared with 104 students in four courses in 1965–66. As these courses develop there is thus an expectation of between 250 and 300 graduates in 1970 and perhaps more in later years. University numbers are not precisely known but from courses now planned 100 or more graduates may emerge annually from 1970 onwards. More than 350 other graduates should have followed a course with a substantial computer science content; this number is open to considerable increase.

These figures are moderately encouraging; together with the estimates for diploma and M.Sc. courses they suggest that we can look for an annual output by 1970 of more than 1,200 computer specialists, of whom a fair proportion may be oriented towards systems design. They may be compared with an overall total of 689 honours graduates in mathematics (including joint honours degrees) in 1960, and 1,290 in 1965.

c) *Curriculum Developments*

We should refer briefly to the probable trends in the development of curricula in this field. Reference to the outlines of existing courses given in this *Bulletin* shows that already there is considerable variety, and this is of course desirable. At the same time much thought has been given to what constitutes a broad balanced curriculum in computer science, particularly at undergraduate level, as shown for instance by the US report[3] published in 1965. The draft syllabuses prepared by the BCS working party also represent a preliminary stage in developing a similar scheme suited to the needs of this country.

It may be several years before syllabuses such as these achieve any general measure of acceptance or stability. In the meantime it is reasonable to expect that first degree courses will develop in at least two directions. One of these will be mathematically oriented, with emphasis on mathematical logic, topology, graph theory, combinatorics, matrix algebra and functional analysis, and will appeal to many who will subsequently become advanced systems programmers. The other, more suited to the systems designer, will be concerned with the theory of economic or engineering systems, and the techniques of information processing in business and administration.

At postgraduate level a different divergence must be expected. There should remain, as at present, broadly based but not too detailed courses for graduates from other disciplines, physics, engineering, economics, medical sciences, etc. However, as first degree courses become more standardized there will also be a need for specialized courses, more experimental in character and closer to the advancing technology of the day.

d) *Ancillary Courses and Equipment*

So far we have mainly been concerned with courses intended to provide computer specialists of one kind or another. It is also becoming essential, as far as teaching staff and computer resources allow, to give all students in the faculties of science, engineering and economics at least a brief introduction to problem formulation and programming in a language such as ALGOL or FORTRAN, and an opportunity for some practical experience.

The practical work represents a potentially large load on computing facilities. To take the *University of London* as a major example, with a total population of more than 30,000 students, there are between 4,000 and 6,000 students each year who should be introduced to computer techniques. (At present less than 1,000 undergraduates annually receive such an introduction, often very restricted.) To provide adequate facilities for them requires the full time equivalent of an IBM 7094, and ideally much of this should be in the form of conversational access, at present non-existent in London. Rapid turn-round of work is in any event essential. There is a need for considerable reorientation of computing services in the universities for teaching purposes.

Further Education

We should now look at the way colleges of further education can exercise their important functions of meeting the technical needs of computer users. For the great majority of colleges this lies in the broader education of programmers, operators and maintenance staff, to use the categories of the inter-departmental report. Reference has already been made to the rôle of computer manufacturers in this respect; undoubtedly the training courses which they provide are very important for the rapid supply of large numbers of programming and operating staff, who are then able to apply particular equipment to solve the more immediate problems of the user. We should also not overlook their usefulness as a means of selecting suitable staff.

However such courses are deficient in at least two respects. First they cannot guarantee that the trainee understands the more general and theoretical aspects of the working and applicability of computers; secondly they do not alone provide a professional qualification which will be acceptable to employer and employee alike. This implies no disrespect to manufacturers' courses, for in the main it is a responsibility of the public educational system to offer these things.

We should distinguish four ways, all of them important and relevant to the computer field, in which colleges can meet existing needs:

1 Short appreciation courses at various levels.
2 Courses designed to provide the knowledge and skills required for particular well-specified jobs.
3 Courses with a wide educational background as well as training in a variety of techniques.
4 Courses which supplement the professional qualifications of students in other fields.

Not all these needs are equally well catered for at present. Appreciation courses there have been and still are in plenty; indeed most college courses have developed out of them. However what now constitutes a satisfactory appreciation course is very different from what it was a few years ago; they require to be carefully designed and illustrated to suit particular classes, from the highest levels of managers to school-leavers in search of a career. Many colleges and other organizations are engaged in providing these, either on their own or as part of a certificated course, as the Society's course list will testify. We shall consider here in more detail some of the more extended courses leading to special qualifications.

The City and Guilds Scheme 319/320: One scheme which has aimed to provide a broad educational background in computer systems is that sponsored by the *City and Guilds of London Institute*. The initiative for this came from the *British Computer Society* in 1962 when an approach was made to the Institute to design courses for junior computer personnel. This led first of all to course 319, basically a two-year part-time course (although other methods of organizing it are possible), which is suitable for those leaving school at 16 or 17 and starting on a computer-oriented career. The award is a certificate for computer personnel. Courses for this began in 1964, and the first examination held in 1966 with 62 candidates. This number should be well over 200 in 1967 and 500 in 1968, with between 50 and 60 colleges offering the course.

The second stage of the scheme, represented by course 320, provides an advanced certificate after two further years of part-time study. Ten colleges at present offer the course and the first examination will be in 1968.

These courses are not designed for a narrow vocational training to meet the needs of a particular class, be it operator, programmer or systems analyst, but rather to offer a core of knowledge and practical experience of computer systems, their mode of operation, method of use and breadth of application. This should appeal to many who, through necessity or otherwise, prefer the path of apprenticeship and further education to a more academic course. Given a serious intention to develop a career in the computer field, the advanced certificate with suitable additional courses can be one avenue to the more responsible positions of computer supervisor, senior programmer or even systems analyst.

So far no form of specialization has been introduced into this City and Guilds scheme, on the grounds that there is more than sufficient common knowledge which is basic to industrial and commercial, as well as scientific and technical uses of computers to fill a course at the level of 319, and even 320. However, the 319 syllabus in particular has provided much just criticism that the target of a general purpose course has not been achieved. For this and other reasons the 319 syllabus is already being subjected to thorough revision and closer integration with that of 320, and the results will be made known before the end of 1967.

National Certificate and other courses: For the technician who is likely to be concerned with the more scientific or technical uses of computers there are other valuable courses available which provide certificates. Some of these are college diploma courses, but of more potential significance are the new Higher National Certificate and Diploma in mathematics, statistics and computing, now under the aegis of the *Institute of Mathematics and its Applications*. These awards parallel a National Certificate scheme (at ordinary and higher levels) which has been operating on a small scale in Scotland for a few years.

Some HNC and HND courses in business studies also now have a significant computer content, although not in any sense aiming to produce computer specialists. For those aiming to become systems analysts there is an important development now in the planning stage between a number of colleges and industry to provide courses of four to six months in college together with industrial training. This is also available in Scotland, where the *Scottish Council for Commercial, Administrative and Professional Education* (SCCAPE) offers a diploma in systems analysis and design for a course on similar lines. The *National Computing Centre* is also concerned to promote courses for systems analysts and is designing a course for the purpose.

These courses are clearly aimed at reorienting many people who may already have professional qualifications in another field, in a direction where there is a severe national shortage. There is a similar need in data processing applications generally, especially for some intermediate levels of managers who find that computers are impinging on the work of their departments. This has led to a further collaboration between the *British Computer Society* and the *City and Guilds of London Institute* to prepare a course on 'data processing for computer users' (course 383), intended to be run as a part-time evening course with about 150 hours of study and practical work. A certificate will be awarded and it is hoped that the first examination for this will be held in 1968.

This short summary has covered the majority of courses extending to 150 hours or more. There is an important place for other short 'supplementary' courses; two of these, entitled 'computer studies I' and 'computer studies II' are offered at some colleges in Scotland. Each involves about 80 hours, and may be taken separately as or part of the Scottish Advanced National Certificate in Business Studies; they are rather more than appreciation courses but not specialist courses for programmers.

Some gaps still remain. Thus computer operators, and those operating ancillary machines, do not as yet have training awards specifically for their needs. This is being remedied both by SCCAPE, which is planning a course similar to one for business machine operators, and by the *Royal Society of Arts*, which has not so far entered the field of computer education. Another region of very great importance is that of computerized automation, which involves among others the maintenance technician. This is now under review by a working party of the *United Kingdom Automation Council*, and is likely to result in modification of several existing courses for technicians as well as possible new courses.

There is a clear danger that all this varied development of courses may lead to inefficiency and to an unrelated mass of disjointed or overlapping qualifications. Thus in the technical field a strong case arises for coordinating the activities of at least three examining bodies, CGLI, RSA and SCCAPE. To these may be added the *British Computer Society* itself. This is a possibility which is certainly being considered with a view to establishing standards which will be generally acceptable.

Computing Equipment in Universities and Colleges

No review of progress in computer education would be satisfactory without reference to the equipment available to students. In universities digital computers have been provided primarily for research, beginning with pioneer machines in Manchester, Cambridge and London about 1949–50, followed by a number of commercially built machines in the period 1956–58, and again in 1962–64. The total government contribution up to this stage was of the order of £3 million. Following the publication of the Flowers report [4] and the setting up of the *Computer Board*, the provision of computers in universities has entered a phase of more systematic planning, with the prospect of a further investment of about £18 million between 1966 and 1972 in machines to provide general computing services.

There are at present about 50 computers in universities in the UK for such services. Although the position is changing fairly rapidly, the numbers and types of machine installed are indicated by the following table:

Ferranti Mercury	1	ICT Atlas	3
Ferranti Pegasus	2	ICT 1907*	1
Ferranti Sirius	3	ICT 1905	2
Ferranti Argus 104	2	ICT 1909	3
Stantec Zebra	4	EE/KDF9	8
Elliott 803	12	Elliott 503	2
IBM 1620	5	Elliott 4100	3
		IBM 360/65	1
		IBM 7090/1401	1*
		IBM 1440	1

The machines on the left include those most likely to be replaced in the near future; at least one further 1905E, one 1907 and several machines in the Elliott 4100 range are on order. Reference should also be made to the SRC Chilton Atlas which is available to universities, and to the equipping of the Edinburgh Regional Centre, first with KDF9 and then with System 4/75. Many small computers exist in departments for specific projects, and several larger machines dedicated to high energy physics.

It has been said that these computers are primarily for research. Until now it is true that the time directly devoted to teaching courses has been relatively small (although two-three hours per week on the London Atlas is not trivial), but a much greater proportion has effectively been providing self-education in their use to a large number of graduate students. This situation will change rapidly as more undergraduate courses get under way.

*Due for upgrading to 7094/1460

In colleges of further education the provision of computers has been more directly related to teaching needs. The following list shows an even greater variety, despite being incomplete at the time of writing:

Ferranti Mercury	1	Stantec Zebra	3
Ferranti Pegasus	3	Elliott 402	1
Ferranti Sirius	2	Elliott 803, 803B	9
EE Deuce	1	Elliott 903	1
ICT 1201	2	Elliott 4100	5
ICT 1202	4	IBM 1620	4
ICT 1301	1	IBM 1130	3
ICT 1901	1	IBM 1401	1
		IBM 1440	2
Honeywell H120	1	PDP 120	1
Honeywell H200	1	MCS 920	1

There are also one or two locally designed computers, *e.g.* NHECTA II at Letchworth. The vintage quality of many of these machines is evident, and a great deal needs to be done to extend and update the equipment in colleges. Neither in universities nor colleges are there as yet any substantial multi-access facilities to enable students to use computers in a conversational mode, although a few, very few, experimental schemes are under way. This is particularly unfortunate as the potential value of such facilities is immense, even if restricted to program editing rather than full conversational use.

Conclusions

It has not been a purpose of this article to present a comprehensive plan for computer education in this country, or even to highlight too strongly the areas of deficiency. Some of the problems involved are discussed elsewhere. I have been more concerned to describe what has been done than to stress what remains to be done. By comparison with five years ago a great deal has been achieved both in universities and colleges of further education, largely through the energy and enthusiasm of those who are aware of the needs of the future and also have the responsibility of presenting the courses. However we are still in the process of developing an integrated system of computer education which will embrace schools as well as institutions of higher education, and will set the requisite standards for professional as well as academic education. Only if continued government support is forthcoming are the targets of 1970 likely to be achieved. Next year's review will be critical.

References

1. *Computer Education: Report of an Interdepartmental Working Group* HMSO, 1967.
2. *Ministry of Labour Manpower Studies No. 4: Computers in Offices* HMSO, 1965
3. *An undergraduate curriculum in computer science* Comm ACM Vol 8, page 543. 1965
4. *Report of a Joint Working Group on Computers for Research* Cmnd 2883 HMSO.

ABOUT THE AUTHOR

Professor R. A. Buckingham is Director of the Institute of Computer Science in the University of London. He was a Reader in Physics at University College before taking charge of the newly formed Computer Unit in 1957 and was made Professor of Computing Science in 1963. He is Chairman of the IFIP Technical Committee on Education and of the City and Guilds Advisory Committee on the Education of Computer Personnel and has held many BCS appointments.

A comparison of courses leading to the whole or a substantial part of a first or higher degree in computer science is presented for the guidance of career masters, students, colleges and anyone wishing to find out the current national position concerning the education of computer scientists.

Survey of computer science courses

This survey is divided into two parts: first degrees and higher degrees. In each part institutions are listed in alphabetical order of the main word in the address, *e.g.* the name of the town.

Each entry consists of eight sections. Section 1 gives the name and address of the department and college. Section 2 gives the name of the qualification: this is usually a B.Sc. for a first degree. The B.Sc. is usually awarded by a university, but, in the cases of colleges of technology, this is awarded by the *Council for National Academic Awards* (CNAA). Section 3 gives the entry requirements. For a first degree, the entry requirement is usually a minimum of two 'A' levels in the General Certificate of Education; in practice it is often required to have three 'A' levels and often the pass has to be of a high standard where entry is very competitive.

Section 4 gives the length of the course. This is usually full-time, but for the CNAA degrees it is often a sandwich course spread over a period of four years with alternating periods in college and industry. Section 5 gives general remarks about the objectives of the course.

Section 6 gives main computing facilities available for the course. Section 7 gives present or estimated future number of students taking the course. Section 8 gives a more detailed description of the contents of the course with number of lectures on the right-hand. The number of lectures taken by full-time students is subject to a large variation, between 8 to 25 lectures per week. Generally, stronger students are expected to do a lot of reading on their own and the number of lectures per week is therefore quite low for such students.

FIRST DEGREES

Brighton College of Technology

Department of Computing, Cybernetics and Management, Brighton College of Technology, Moulsecoomb, Brighton 7

B.Sc. (Hons) in computing and data processing

2 GCE 'A' subjects, 1 of which must be mathematics, 3 other GCE 'O' subjects. ONC or equivalent with good performance in mathematics

4 years: 9 terms (11 weeks each) academic, the rest industrial training. There are three different schemes (two thin, one thick sandwich) for combining academic and industrial training

5 Suitable for school leavers who intend to take up computing as a career. Emphasis on commercial and management applications

6 ICT 1301, IBM 1401

7 14 (64/65), 46 (65/66)

8

Term 1	
Mathematics and computing methods	44
Logic, sets, Boolean algebra	16
Programming	44
Logical design	22
Statistics	66
Scientific thinking and method	11
Economics	22
Accounting	22
Term 2	
Mathematics and computing methods	44
Logic, sets, Boolean algebra	16
Programming	44
Logical design	22
Statistics	66
Physics	33
Economics	16
Accounting	16
Term 3 as in Term 2	
Term 4	
Mathematics and computing methods	44
Programming	44
Logical design	22
Statistics	11
Either:	
Scientific thinking and method	22
Analogue computing	22
Electronics and information theory	33
or Psychology and sociology	22
Economics	28
Sociology	28

Term 5 as in Term 4 but with an essay project instead of statistics

Term 6

Mathematics and computing methods	33
Digital computing	33
Statistics	33
Either: Analogue computing	22
or Commercial applications	22
Management organization	22
Technical factors and social change	12
Organization of science	8
Operational research	22
Project essay	11

Term 7

Mathematics and computing methods	44
Digital computing (including project)	77
Statistics	44
Either:	
Scientific applications	11
Analogue computing	22
or Commercial applications	33
Operational research	44

Term 8 as in Term 7

Term 9 as in Term 7

City University

1 Department of Mathematics, City University, St. John St, London, EC1

2 B.Sc

3 'A' level in pure mathematics, applied mathematics and physics (preferably), or the equivalent + English language and 2 other subjects at 'O' level

4 3 years, or 4 years on sandwich basis

5 Suitable for those who wish to specialize on computers for their career

6 ICT 1905, Elliott G-PAC analogue computer which will be replaced by EAL 690, hybrid computer from Sept. 1967

7 The present annual intake is about 40 students

8

Part I (20 weeks)

Mathematics (including numerical analysis)	140
Mechanics	40
Statistics	40
Computing	20
Social studies	30

Part II (40 weeks)

Mathematics (including numerical analysis)	260
Field theory and mechanics	50
Statistics	110
Digital computing	50
Analogue computing	40
Logical design	20

Part III (40 weeks)

Mathematics (including numerical analysis)	120
Digital computing	160
Analogue hybrid computing	80
Logical design	40
Operational research techniques	60
Social studies	80

Constantine College of Technology

1 Department of Mathematics, Statistics and Computing, Constantine College of Technology, Middlesbrough

2 B.Sc (Ordinary) in computer science

3 i) GCE – 5 subjects including mathematics and 1 other subject both at 'A' level. Special consideration will be given to applicants without mathematics at 'A' level; ii) good ONC; iii) qualification equivalent to the foregoing.

4 4 years, including 2 periods of 6 months in industry

5 Suitable for school leavers and others who wish to become advanced or specialist programmers or analysts in commercial and/or technical fields

6 IBM 1620 Model 2; disc packs; line printer

7 21 (1966/67)

8

First year (in common with honours degree)

(Commences with a two-week FORTRAN II course and an introduction to desk-machines – not included in 'number of lectures')

Programming I	15
Principles of computers	70
Basic electronics	20
Data processing I	20
Statistics I	60
Commercial organization and procedures	30
Mathematics I	100
Numerical analysis I	30
Communication	15
Economics	30
Tutorials and practicals	320

Second year

Boolean algebra and logic design	20
Programming II	40
Data processing II	20
Statistics II	40
Mathematics II	40
Numerical analysis II	40
Foreign language	20
Tutorials and practicals	200

Third year

Programming III	30
Data processing III	40
Analogue computing I	20
Operations research	40
Mathematics III	40
Numerical analysis III	20
Accounting for management	20
Psychology	20
Tutorials and practicals	240

Fourth year

Real-time systems	60
Data transmission and information theory	30
System methodology	[illegible]
Programming IV	30
Analogue computing	30
Computer applications	90
Project	—
Industrial sociology	30
Tutorials and practicals	300

Constantine College of Technology

1 Department of Mathematics, Statistics and Computing, Constantine College of Technology, Middlesbrough

2 B.Sc (Hons) in computer science

i) GCE – 5 subjects including mathematics and 1 other subject both at 'A' level. Special consideration will be given to applicants without mathematics at 'A' level; ii) good ONC; iii) qualification equivalent to the foregoing.

4 years, including 2 periods of 6 months in industry

Suitable for school leavers and others who wish to become advanced or specialist programmers or analysts in commercial and/or technical fields

IBM 1620 Model 2; disc packs; line printer

21 (1966/67)

rst-year (in common with ordinary degree)

ommences with a two-week FORTRAN II course and an troduction to desk-machines – not included in 'number of ctures')

ogramming I	15
inciples of computers	70
isic electronics	20
ata processing I	20
atistics I	60
ommercial organization and procedures	30
athematics I	100
umerical analysis I	30
ommunication	15
onomics	30
itorials and practicals	320
cond year (20 weeks)	
ogramming II	20
olean algebra and logic design	20
ialogue computing I	20
ita processing II	20
itistics II	40
athematics II	40
imerical analysis II	40
counting for management	20
nguage laboratory	20
torials and practicals	200
ird year (20 weeks)	
ogramming III	20
al-time processing systems	20
stems methodology	20
ita processing III	30
erations research	40
tion A Management science	80
Analogue computing	10
Numerical analysis III	10
tion B Numerical analysis III	40
Mathematics and fundamentals of control theory I	50
Analogue computing II	20
chology	20
orials and practicals	200
rth year	
nguages and compilers	30
l-time system programming	30
a transmission	30
te state systems	30
ject	—
on A Advanced management systems	
Integrated data processing	45
Operations research	45
ion B Numerical analysis IV	45
Mathematics and fundamentals of control theory II	
Analogue computing III	30
Industrial sociology	30
Tutorials and practicals and project	360

Glasgow University

1 Computing Department, Glasgow University, Glasgow, W2

2 Courses for B.Sc and B.Sc (Applied science)

3 *Higher Ordinary Computing:* A pass in mathematics at the ordinary level
Advanced Ordinary Computing: Passes in computing *and* mathematics at the higher ordinary level

4 3 years for ordinary degree

5 Provides a basic course in computing (higher ordinary class) and course for those wishing to work in computer programming in science and engineering or in computer systems (advanced classes)

6 English-Electric-Leo-Marconi computer KDF 9

7 Estimated – Higher Ordinary (1966/67) 100
Estimated – Advanced Ordinary (1967/68) 25

8 Under the regulations for the degrees in science and applied science in the University of Glasgow, a student has to select seven courses to be studied in three sessions. Computing may be studied in the second and third years and must be preceded by classes in mathematics.

Higher ordinary computing: numerical analysis 100

Difference methods including operators, interpolation, quadrature, and the interpolating polynomial; iterative methods for the solution of transcendental and polynomial equations; introduction to methods of linear algebra; bivariate functions.

Higher ordinary computing: computing

Fundamentals of computers; computer programming for numerical problems; computer logic; information processing.

Advanced ordinary computing: numerical analysis 100

Numerical solution of ordinary differential equations; linear algebra including eigenvalues; ordinary differential equations; linear programming methods; methods of approximation; partial differential equations; non-linear problems.

Advanced ordinary computing: automatic computing

Structure of computer languages; programming in assembly language; computer systems; compilers; non-numerical programming.

Computer systems: 100

Computer design and construction

Order codes (0, 1, 2, 3 address); error detection and correction codes; peripheral hardware; hardware protection.

Computer systems

Requirements of an operating system; batch processing operating environment; Egdon and Prompt operating system; Atlas and CDC 6600 operating systems; multi-access systems – project MAC; directors; systems hardware.

Compilers and system writing

Machine dependence; buffering of input-output; store organization; programming of asynchronous devices; interlocks; debugging techniques in language; graphical and analogue devices.

Hatfield College of Technology

1 Department of Mathematics, Hatfield College of Technology, College Lane, Hatfield, Herts

2 B.Sc (Hons) in computer science

3 GCE 5 subjects, including 2 at 'A' level, one of which must be mathematics, *or* OND *or* ONC in science, engineering or business studies with a mark of *at least* 60 per cent. in mathematics

4 4 years, including 2 6 months' industrial training periods

5 Suitable for school leavers who intend to take up computing as a career

6 Elliott 803

7 14 (1965/66) 52 (1966/67) (Common with ordinary degree in first year)

8

First year (Common with ordinary degree)

Digital computer programming	60
Practicals	120
Electronic data processing I	30
Practicals	30
Computational mathematics, statistics, electronics, and liberal studies, lectures and practicals	450

Second year

Programming languages I	20
Practicals	20
Electronic data processing II	40
Practicals	40
Computational mathematics, statistics, electronics, accounting, liberal studies	300

Third year

Analogue computing I	20
Practicals	40
Programming languages II	20
Practicals	40
Logical design I	30
Practicals	30
Computational mathematics, statistics, management and liberal studies	240

Fourth year

Information transmission	30
Complete systems	45
Analogue computing II (practicals)	45
Theory of automata and systems programming	60
Practicals	60
Either: Logical design of computers II *or* Digital computer applications I (Integrated data processing) *or* Digital computer applications II (Simulation) *or* Digital computer applications III (Information storage and retrieval)	45
Practicals	45
Liberal studies	60
Project	180

Hatfield College of Technology

1 Department of Mathematics, Hatfield College of Technology, College Lane, Hatfield, Herts

2 B.Sc (Ordinary) in computer science

3 GCE 5 subjects, including 2 at 'A' level, one of which must be mathematics, *or* OND *or* ONC in science, engineering or business studies with a mark of *at least* 60 per cent. in mathematics

4 4 years, including 2 6 months' industrial training periods

5 Suitable for school leavers who intend to take up computing as a career

6 Elliott 803

7 52 (1966/67) (Common with honours degree in first year)

8

First year (Common with honours degree)

Digital computer programming	60
Practicals	120
Electronic data processing I	30
Practicals	30
Computational mathematics, statistics, electronics, and liberal studies, lectures and practicals	450

Second year

Computer applications and programming languages I	30
Practicals	50
Electronic data processing II	40
Practicals	40
Computational mathematics, statistics, accounting, liberal studies	260

Third year

Analogue computing I	20
Practicals	40
Computer applications and programming languages II	20
Practicals	60
Logical design	30
Practicals	30
Computational mathematics, management and liberal studies	190

Fourth year

Information transmission	30
Complete systems	45
Analogue computing II (practicals)	30
Integrated data processing	60
Practicals	60
Either: Systems programming *or* Simulation *or* Information storage and retrieval	45
Practicals	45
Liberal studies	60
Project	180

Heriot-Watt University

1 Department of Mathematics, Heriot-Watt University, Chambers Street, Edinburgh

2 B.Sc (Hons) in computer science

3 Scottish Certificate of Education including physics and mathematics or GCE with 'A' levels in physics and mathematics

4 4 years – full-time

5 Suitable for school leavers who intend to take up computing as a career. Emphasis on scientific and mathematical applications

6 ICT Sirius

7 Course commenced in October 1966

8

First year

Pure mathematics	180
Applied mathematics	90
Chemistry	150
Physics	210

Second year

Pure mathematics	180
Algorithms, computers, computer organization, and programming	180

Numerical analysis I	60
Probability and statistics	60
Humanities subject	60
Third year	
Pure mathematics	120
Numerical analysis II	120
Information structures, computer and programming systems	90
Combinatorics and graph theory	60
Principles of analogue computation	60
Special physics course	60
Humanities subject	60
Fourth year	
Mathematical techniques of science and engineering	100
Numerical analysis III	125
Algorithmic languages and compilers	75
Logical design and switching theory	75
A special topic to be chosen from:	
Mathematical optimization techniques	60
Advanced analogue and hybrid computation	60
Commercial data processing	60
Constructive logic and theory of automata	60
Systems simulations	60
Formal languages	60
Heuristic programming	60

University of Lancaster

1 Department of Mathematics, University of Lancaster

2 B.A

3 Usually 3 'A' levels, including mathematics, at about 'B' standard

4 3 years

5 The general aim of the course is to produce mathematicians who have both a substantial grounding in the theoretical aspect of the subject and some knowledge of *modern* applications, such as statistics, OR, numerical analysis, computer science, etc.

6 Until January 1967: IBM 1620 with disc
From December 1966: ICT 1909

7 Next term: 3rd year: 7; 2nd year: 14; 1st year (including students majoring in other subjects): 150

8

First year	
Introduction to computing	10
Practicals	10
Second year	
Elementary theory of computation, machine codes, logic design	30
Introduction to some special-purpose languages	
Numerical analysis	30
Third year	
Compilers	10
LP languages, on-line and other applications	15
Numerical analysis	20

University of Leeds

1 Computing Laboratory, University of Leeds, Leeds 2

2 B.Sc (Combined Honours) in computational science and mathematics or computational science and physics

3 Good passes in 3 subjects at the advanced level of the GCE of which mathematics (pure and applied) must be included

4 3 years full-time

5 Suitable for undergraduates who wish to take a broadly based honours degree. About 75 per cent. of the time is divided equally between the two principal subjects and the remainder on the subsidiary subject (mathematics in the case of the computational science/physics combination)

6 KDF 9, Elliott 903 and some analogue facilities

7 40 (1967/68)

8

First year

Six hours per week devoted to ALGOL programming, computing systems, numerical analysis, algebra and practical and seminar classes.

Second year

Six hours per week devoted to information processing, theory of programming languages, operational research, numerical analysis and practical and seminar classes.

Third year

Six hours per week devoted to mathematical models, operational research, numerical analysis and a special topic together with practical and seminar classes.

University of Leeds

1 Computing Laboratory, University of Leeds, Leeds 2

2 Honours B.A or B.Sc in mathematics (alternative scheme of study)

3 Good passes in mathematics, including pure and applied, either as 1 subject or 2 subjects at 'A' level

4 3 or 4 years full-time, depending on the standard at entry

5 Suitable for undergraduates who wish to specialize in the final honours year in computational mathematics and mathematical statistics. Undergraduates register initially for the B.A or B.Sc in mathematics and decide during the course on the scheme of study for the final year

6 KDF9, Elliott 903 and some analogue facilities

7 Available from October, 1967 with a maximum of 25 students for the first year

8 11 hours per week plus practical and seminar classes. Two hours per week is spent on basic courses in each of computational mathematics, mathematical statistics and mathematical methods. The remaining five hours per week are devoted to optional courses in some or all of these topics.

Leicester Regional College of Technology

1 Department of Mathematics, Computing and Statistics, Leicester Regional College of Technology, The Newarke, Leicester

2 B.Sc in computer science

3 GCE in 5 subjects including 2 at 'A' level one of which must be mathematics and including physics at 'O' level; or good ONC

4 4 years including 2 6 months' industrial training periods

5 Suitable for school leavers who intend to take up computing as a career

6 Honeywell 200

7 24 (estimate)

8

First year	
Programming fundamentals	96
Mathematics	96
Physics and electronics	96
Numerical analysis	64

Data processing	64
Tutorials, laboratory classes + liberal studies	288
Second year + third year (Industrial training takes place during these years)	
Programming fundamentals	99
Mathematics	66
Physics and electronics	66
Numerical analysis	66
Statistics + operational research	51
Data processing	66
Tutorials, etc.	264
Fourth year	
Advanced systems programming	96
Information processing theory	64
Lists, heuristic programming + artificial intelligence	64
Project	192
Tutorials, etc.	320

University of Liverpool

1 The Department of Computational and Statistical Science, University of Liverpool, Liverpool 3

2 B.Sc

3 3 'A' levels

4 3 or possibly 4 years

5 The course is designed to give scientists (mainly mathematicians and physicists) a knowledge of computational and statistical science

6 KDF9 computer

7 20 (1966/67)

8 The course below will usually form about 40 per cent. of the student's load in his second year.

Statistics	100
Numerical analysis	60
Automatic computation	80

University of Liverpool

1 The Department of Computational and Statistical Science, University of Liverpool, Liverpool 3

2 B.Sc in science with honours in mathematics

3 3 'A' levels

4 3 or possibly 4 years

5 Gives a main training in mathematics with subsidiary physics

6 KDF9 computer

7 38 honours mathematics students. Up to 25 taking the options below (1965/6)

8

Numerical analysis	40
Mathematics of operations research	40
Approximation theory	40
Automatic computation	40

In the final year of the degree, students must take 6 courses from a total of about 12. The choice includes the above.

University of London: Birkbeck College

1 Department of Computer Science, Birkbeck College, Malet Street, London, WC1

2 B.Sc

3 3 'A' levels

4 3 years part-time

5 A degree in mathematical sciences with a substantial computer science content, may be taken part-time

6 Atlas, ICT 1400

7 30 taking all options; a larger number will take the earlier courses.

8 Under the new science degree structure, students can choose a wide variety of courses from various subjects provided the total number of courses add up to about 800 lectures. The choice includes the following:

Computer science 1 (a basic course including elementary programming)	80
Computer science 2 (more advanced programming techniques)	80
Numerical analysis 1 (introduction to numerical methods)	80
Numerical analysis 2 (more advanced numerical methods in linear algebra and partial differential equations)	80

University of London: Queen Mary College

1 Department of Mathematics, Queen Mary College, Mile End Road, London, E1

2 B.Sc

3 2 'A' levels

4 3 years

5 Suitable for school leavers who wish to get an all-round mathematics degree specializing in their third year in computer science, statistics, pure mathematics, or applied mathematics

6 ICT 1905E and on-line data link to the London Atlas

7 60 (1967/68), 60 (1968/69)

8

First year	
Introduction to computer science	40
Statistics, pure and applied mathematics	240
Second year	
Numerical methods	40
Basic computer science	80
Selection from statistics, pure and applied mathematics	160
Third year	
More advanced topics in computer science	120
Numerical analysis	40
A selection from statistics, pure and applied mathematics	80

University of Manchester

1 Department of Computer Science, The University, Manchester 13

2 B.Sc (Hons) in computer science

3 3 'A' levels including mathematics and physics

4 3-year course

5 Emphasis on computer engineering, advanced programming and computer systems

6 Manchester Atlas

7 Intake = 40

8

First year	
Systems logical design	60
Numerical analysis and programming	30
Programming practical	120
Mathematics and physics lectures, practicals and tutorials	400
Second year	
Memory systems and basic computer circuits	60
Semiconductor electronics	90
Electronics laboratory	150
Numerical analysis	60
Programming	60
Programming practical	120

Applications 30
Mathematics lectures and tutorials 120

Third year

Systems programming 90
Applications 30
Large scale systems design 30
Two out of the following:
Computer circuits and advanced devices 90
Control electronics and circuit theory 90
Mathematical methods 90
Numerical analysis 90
Project 4 afternoons a week for 24 weeks

University of Newcastle upon Tyne

1 Computing Laboratory, 1-3 Kensington Terrace, Newcastle upon Tyne 2
2 B.Sc in computing science
3 3 'A' levels (including mathematics)
4 3 years
5 Emphasis on numerical and non-numerical applications of computers, programming and appropriate mathematical topics
6 KDF9; IBM 360/67
7 Initially 12 – increasing from 1968/69
8

First year

Mathematical ground work similar to students entering special honours mathematics or physics degrees.

Second and third year

About 600 hours of lectures and tutorials on: programming; numerical analysis; data handling; systems analysis; mathematical statistics; operations research; logical and systems design.

University of Newcastle upon Tyne

1 Computing Laboratory, 1-3 Kensington Terrace, Newcastle upon Tyne 2
2 B.Sc general degree with honours
3 3 'A' levels (including mathematics)
4 3 years
5 Automatic computing forms one quarter of the course. The remainder of the course can be pure mathematics and either applied mathematics and statistics, or another science subject such as physics.
6 KDF9; IBM 360/67
7 60
8 About 100 hours of lectures and 100 hours of class practicals spread over two years on: programming; numerical analysis; combinatorial and non-numerical problems.

University of Reading

1 Computer Unit, University of Reading, Reading, Berks
2 B.Sc (Hons)
3 'A'-level in pure mathematics, applied mathematics and (normally) one other subject
4 3 years
5 Leads to a three-subject honours degree in pure mathematics, applied mathematics, and computer science. (For the first six months of the course the student must read another subject, *e.g.* physics, in place of computer science)
6 Elliott 4130 with magnetic tapes (as from August 1967)
7 20 – 40 each year
8

Pure mathematics 280
Applied mathematics (including statistics and numerical analysis) 280
One other subject (first year only) 80
Computer science: 200
Programming languages
Programming techniques
Compilers
Logical design of computers
Supervisor and test programs
Numerical and non-numerical data processing
Other applications of computers

Staffordshire College of Technology

1 Department of Science, Mathematics and Computing, Staffordshire College of Technology, Beaconside, Stafford
2 B.Sc in computing science
3 Any 2 GCE 'A' subjects, 3 other GCE 'O' subjects
4 4 years, the third year is industrial training
5 Suitable for school leavers who intend to take up computing as a career. Supported by the nearby English-Electric computer firm
6 Deuce in the college. Access to EELM bureau machines at Kidsgrove
7 11 (1965/66), 60 (1966/67)
8

First year

Computer systems and logic 96
Information theory and systems 57
Theory and principles of programming 57
Applications programming 64
Mathematics, basic science, economics, statistics, liberal studies 548

Second year

Computer systems 69
Information systems 246
Theory and principles of programming 169
Applications programming 92
Mathematics, statistics, liberal studies, economics 250

Third year

Industrial period

Fourth year

Computer systems 44
Information systems 88
Theory and principles of programming 88
Either: Simulation theory
or Statistics
or Mathematical programming
or Non-numeric applications 132
Mathematics, liberal studies 220
Project 264
Three options from the following:
Calculus of variations, artificial intelligence, linear algebra, optimization techniques, hybrid computers, data transmission and control theory, data processing in business problems, operational research, differential equations, machine tool control by computer 132

University of St Andrews

1 Mathematics, Physics and Computing Laboratory, University of St Andrews, St Andrews, Fife

2 B.Sc in computational science

3 2 'A' levels including mathematics. At least 1 other 'O' level, in physics if this has not been passed at 'A' level

4 4 years honours, 3 years ordinary

5 First year course at general level
Second year course at special level
Third and fourth years at honours level

6 IBM 1620 Model 2, with discs, printer and digital plotter

7 Course starts 1966/67. No special restrictions on numbers

8

First year

Electricity and magnetism	18
Statistics	40
Electronics	18
Computer applications	22

Candidates must also attend first year courses in mathematics and applied mathematics.

Second year

Numerical analysis	18
Boolean algebra and logical design	18
Statistics and operational research	18
Programming languages	18
Data processing	18
Switching design	12

Third year and fourth year

Commencing in October 1968. Honours level for candidates who have attended first and second years.

Wolverhampton and Staffordshire College of Technology

1 Department of Mathematics and Physics, Wolverhampton and Staffordshire College of Technology, Wulfruna Street, Wolverhampton

2 B.Sc honours in computer science

3 GCE 5 subjects, 2 at least at 'A' level, including mathematics, physics or chemistry; good ONC; or equivalent qualification

4 4 years, including 3 6 months' industrial training periods

5 Suitable for school leavers who intend to take up computing as a career

6 IBM 1620, Redifon 10/20 analogue computer

7 32 (1965/66)

8

First year

Programming principles	64
Numerical analysis	64
Commercial procedure and applications	128
Analogue computing	48
Mathematics, statistics, physics, liberal studies	432

Second year

Programming principles	44
Numerical analysis	44
Either:	
Commercial procedure and applications	88
or Analogue computing	44
Physics	44
Circuit theory, electronics, computer engineering	66
Mathematics, statistics, liberal studies	253

Third year

Programming principles	104
Liberal studies	52
Either:	
Commercial procedure and applications	260
Statistics and operational research	156
or Statistics and operational research	156
Numerical analysis	52
Pure mathematics	78
Scientific applications	156
or Electrical engineering	104
Applied electronics	104
Computer engineering	104
Information theory and terminal equipment	91
Production processes	52

Fourth year

Programming principles and computer studies	156
Project	156
Liberal studies	52
Either:	
Commercial procedure and applications	104
or Numerical analysis	104
or Advanced electronics and line communication	
Design theory	156

HIGHER DEGREES

University of Bradford

1 Computing Laboratory, The University of Bradford, Bradford 7

2 M.Sc computer science

3 Basically a diploma of technology or bachelor degree with honours in mathematics, statistics, physics, chemical engineering, civil engineering, mechanical engineering

4 1 calendar year

5 To provide postgraduate training in computers and applications. Particular reference to process control and analogue and hybrid techniques is a feature of the course

6 Stantec computer, ICT 1909 computer, Emiac analogue computer

7 6 – 12 per annum

8

History and evolution of computers and computing techniques	5
Theory and logic of computation	50
Computer software	100
Numerical analysis	100
Computer hardware	50
Analogue and hybrid systems	100
Applications of computers	50
	450
Plus tutorials	50
	500
Project	150
	650

16 hour/week for 40 weeks

University of Cambridge

1 Mathematical Laboratory, Corn Exchange Street, Cambridge

2 Diploma in computer science

3 A good honours degree in a scientific subject

4 9 months full-time

5 Suitable for graduates intending to take up research and development in the computer industry or universities

6 Titan and PDP 7

7 20

8

First term

Computer design, programming, software, numerical analysis	40

Second and third term

Either:

	Symbol manipulation, compilers, operating systems, programming languages, list processing	40
or	Numerical methods for partial differential equations, numerical methods for matrices, advanced finite difference theory, orthogonal polynomials	
or	The design of computer systems, logic design, error-correcting codes, relevant topics in mathematical logic, linear programming	

A dissertation

City University

1 Department of Mathematics, City University, St John St, London, EC1

2 M.Sc

3 An honours degree (normally 1st or 2nd class in mathematics)

4 Maximum of 3 years, part-time, 1 day per week. (The lecture course of 2 years + a dissertation submitted subsequently)

5 The course is designed for postgraduate students who wish to specialize in the application of computers to technical problems, and in the logic design of computer systems including both hardware and basic software.

6 ICT 1905. Elliott G-PAC analogue computer which will be replaced by EAL 690/hybrid computer from September 1967

7 Present intake 6 per year

8

First year

Numerical analysis	
Linear algebra	30
Theory of approximation	15
Ordinary differential equations	25
Digital computing	
Problem oriented languages	15
Logical design of computer systems	20
Analogue-hybrid computing	
Basic analogue techniques	10
Parallel hybrid computers	15
Subservient digital applications	10

Second year

1 out of the following 3 options:

1 Numerical analysis	
Partial differential equations	25
Integral equations	15
Optimization techniques	30
2 Digital computing	
Theory of programming	45
Non-numerical applications	15
Large computer systems	10
3 Analogue-hybrid computing	
Full hybrid systems	35
Advanced analogue techniques	10
Error analysis, stability and optimization	10
Field problems	5
Sampled data systems and random processes	5
Discrete simulation	5

Cranfield, The College of Aeronautics

1 Department of Mathematics, The College of Aeronautics, Cranfield, Bedford

2 Diploma in the application of computers to technological problems

3 A good honours degree or equivalent professional qualification

4 1 year full-time

5 Main emphasis is on the use of computers in engineering. Suitable for engineers, senior programmers, systems analysts and system designers

6 ICT 1905. Applied Dynamics AD256 iterative analogue

7 Restricted to 10 in 1967/68

8

Numerical analysis	30
Programming languages	30
Logical and circuit design	30
Digital and analogue computers	30
A problem orientated language	30
A technological specialization	
A computer project	
A dissertation	

Cranfield, The College of Aeronautics

1 Department of Mathematics, The College of Aeronautics, Cranfield, Bedford

2 Diploma in the application of computers to business and management problems

3 A good honours degree or equivalent professional qualification

4 1 year full-time

5 Main emphasis is on the use of computers in management. Suitable for senior programmers, systems analysts and designers

6 ICT 1905. Applied Dynamics AD256 iterative analogue

7 Restricted to 10 in 1967/68

8

Numerical methods	30
Programming languages	30
Probability and statistics	30
Business and management systems	30
A problem orientated language	30
A business or management specialization	
A computer project	
A dissertation	

University of Edinburgh

1 Department of Machine Intelligence and Perception, University of Edinburgh, Hope Park Square, Meadow Lane, Edinburgh 8

2 Diploma in machine intelligence and perception

3 Honours degree

4 1 year full-time

5 To collect together into one coherent course, those parts of those subjects which seem to be relevant for those who wish to do research in the area of machine intelligence and perception studies.

6 NRC Elliott 4100

7

8

Principles of programming in ALGOL 60	10
Principles of programming in POP-2	10
Theory of programming	18
Software and operating systems	14
Elementary statistics, probability and classification methods	10
Information theory and applications	5
Combinatorics and graph theory	10
Sequential decision-taking and trial-and-error learning	5
Heuristic problem-solving	10
Theorem-proving and fact retrieval programs	10
Pattern recognition	5
Game-playing and game-learning automata	5
Graphical display techniques	10
The processing of visual information by brains and machines	10

Glasgow University

1 Computing Department, Glasgow University, Glasgow, W2

2 Diploma in computing science

3 Degree with mathematics and basic computing, and numerical analysis

4 One academic year – full-time

5 Suitable for graduates wishing to take a course in computer programming or in computer systems

6 English-Electric-Leo-Marconi KDF9 computer

7 Present (1966/67) 15 – 20

8

Automatic computing 60

Basic machine language; syntax and semantics of computer language; data structures; compilers; files and file handling.

and either

Numerical analysis 60

Latent roots and vectors; ordinary and partial differential equations; linear programming and other methods of optimization.

or

Computer systems 60

Computer design and construction; operating systems; system hardware; compiler writing; system programming.

Examination

Two written and one practical paper. In addition candidates submit a dissertation and are examined orally.

University of Leeds

1 Computing Laboratory, University of Leeds, Leeds 2

2 M.Sc by examination in electronic computation

3 Good honours degree in mathematics or in a pure or applied science

4 1 calendar year full-time

5 Suitable for graduates who wish to receive a broadly based training in computing systems, programming and applications

6 KDF9, Elliott 903 and some analogue facilities

7 About 15 students

8

KDF9 system and programming	30
Computing systems	30
Basic numerical analysis	40
Data processing techniques	12

Plus 30 lectures selected from short courses in computational statistics, operational research, numerical analysis, logical design and symbol manipulation.

Plus seminars, tutorials and practical work.

The last 30 per cent. of the course is devoted to project work, which is written up as a short dissertation.

University of Liverpool

1 Department of Computational and Statistical Science, University of Liverpool, Liverpool 3

2 M.Sc in numerical analysis and electronic computation

3 A good first degree in mathematics or science

4 1 calendar year

5 Main emphasis is on the use of computers. Suitable for graduates intending to take up research and development in the computer industry and universities

6 KDF9 computer

7 6 (1965/66)

8

Numerical analysis	100
Operations research	60
Electronic computation	80

A dissertation must be written during the summer.

University of London: Birkbeck College

1 Department of Computer Science, Birkbeck College, Malet Street, London, WC1

2 M.Sc in computer science

3 A good first degree

4 2 years, part-time evening. (Or 1 year full-time)

5 Suitable for graduates intending to take up research and development in the computer industry or universities, for lecturers at colleges of technology hoping to specialize in computer science, and so on

6 Atlas

7 35

8

Basic computer science 40

Two of the following:

Design of programming systems; theory of computation, automata and algorithmic languages; numerical methods; numerical analysis; mathematical optimization; computer methods in operational research; information processing and statistical methods 80

Project

University of London: Imperial College

1 Centre for Computing and Automation, Imperial College, Royal School of Mines Building, Prince Consort Road, London, SW7

2 Diploma of Imperial College

3 A good first degree

4 1 calendar year

5 Suitable for graduates intending to take up research and development in the computer industry or universities

6 Atlas, IBM 7094

7 15

8

Introduction to FORTRAN programming	10
Digital computer organization and design	10
Machine code programming	15
Programming languages	20
Introduction to mathematical methods in computing	20
Data structures and processing	15
A selection of about 60 lectures from the following:	
Business and industrial applications	20
Computing systems	20
Formal languages and compilers	15
Algebraic machine and language theory	10
Advanced numerical methods	20
Data processing in the humanities	5
Graphical data processing	5
Computer logic and circuits	20
Automatic learning and pattern recognition, heuristic methods of problem solving	10
Simulation theory	10

University of London: Institute of Computer Science

1 Institute of Computer Science, 44 Gordon Square, London, WC1

2 M.Sc in computer science

3 A good first degree (normally first or second class honours)

4 1 calendar year

5 Suitable for graduates intending to take up research and development in the computer industry or universities, or to teach computer science in technical colleges or elsewhere

6 Atlas

7 25 full-time; 15 part-time

8

Basic computer science	40
Two of the following:	
Design of programming systems; theory of computation, automata and algorithmic languages; numerical methods; numerical analysis; mathematical optimization; computer methods in operational research; information processing and statistical methods	80

In addition to attending the lecture courses students will spend a considerable number of hours on tutorials and practical work.

Project. This must be written up for submission to the examiners and may also include practical work.

University of Manchester: Institute of Science and Technology

1 Department of Mathematics (Computation), University of Manchester, Institute of Science and Technology, PO Box 88, Sackville Street, Manchester 1

2 M.Sc in automatic computation

3 A good first degree in science or engineering

4 12 months full-time

5 Suitable for scientists and engineers interested in the application of digital computers

6 University of Manchester ATLAS computer, Elliott 803, Ferranti ARGUS

7 About 8 per annum

8

Basic mathematics	48
Preliminary numerical analysis	24
Statistics	24
Approximation and representation	24
Ordinary differential equations	24
Numerical linear algebra	12
Partial differential equations	24
Non-linear problems	16
Programming in Atlas Autocode	10
Practical work and tutorials	

If required for course work:

Programming for the Elliott 803 computer

Programming for the Argus

Examination in May followed by project work for dissertation

University of Newcastle upon Tyne

1 Computing Laboratory, 1-3 Kensington Terrace, Newcastle upon Tyne 2

2 M.Sc in numerical analysis and automatic computing

3 A good honours degree in a scientific subject ensuring an adequate knowledge of mathematics

4 12 months full-time

5 Suitable for graduates intending to take up research and development in computer applications in industry or universities

6 KDF9; IBM 360/67

7 About 7 per annum

8

Lectures on:

Programming; numerical analysis; operational research; logical design; non-numerical applications 200

Extensive practical work

A dissertation

University of Newcastle upon Tyne

1 Computing Laboratory, 1-3 Kensington Terrace, Newcastle upon Tyne 2

2 Diploma in numerical analysis and automatic computing

3 A degree in a scientific subject guaranteeing an adequate standard in mathematics

4 9 months full-time

5 Suitable for graduates intending to apply computers in industry and commerce

6 KDF9; IBM 360/67

7 8 – 10 per annum

8

Lectures on:

Programming; numerical analysis; operational research; logical design; non-numerical applications 200

Extensive practical work

University of Newcastle upon Tyne

1 Computing Laboratory, 1-3 Kensington Terrace, Newcastle upon Tyne 2

2 Diploma in data processing in business administration

3 A good degree or, exceptionally, equivalent qualifications

4 9 months full-time

5 Suitable for those intending to apply computers in industry or commerce

6 KDF9; IBM 360/67

7 4 – 6 per annum

8

Lectures on:

Accounting; business administration; business economics and	50
Programming; systems design; computing equipment; operations research; case studies of commercial applications; non-numerical problems.	100

Extensive practical work.

University of Nottingham

1 Cripps Computing Centre and Department of Mathematics, University Park, Nottingham

2 M.Sc

3 Good degree

4 1 year, or 2 years part-time

5 Computing forms one option. Students choose two. Other half can be statistics, mathematical logic, or mathematical methods

6 KDF9

7 10

8

Mathematical logic
ALGOL programming
Numerical analysis
Non-numeric applications
Some compiler writing techniques
Advanced projects organized by the Cripps computing centre.

Queen's University of Belfast

1 Department of Computer Science, Queen's University, Belfast 7

2 Diploma in numerical mathematics and computing science

3 Pass degree

4 10 months

5 Produce technical staff for research and development in computing

6 ICT 1905

7 10

8

Design and structure of computers	10
Machine coding, assembler, I/O systems, etc.	10
Higher level languages	20
Basic programming	3
Numerical mathematics	40
Statistics	12

Queen's University of Belfast

1 Department of Computer Science and Applied Mathematics, Queen's University, Belfast 7

2 M.Sc in numerical mathematics and computing science

3 Good honours degree

4 1 year

5 For persons intending to do research and/or development in computing. Prerequisite for Ph.D program

6 ICT 1905

7 9

8

Design and structure of computers	10
Machine coding, assembler, I/O systems, etc.	10
Higher level languages	20
Basic programming	3
Numerical mathematics	40
Statistics	12
Advanced programming, list structures, etc.	15
Special topics in numerical analysis	15
Advanced statistics	15

University of St Andrews

1 Computing Laboratory, University of St Andrews, St Andrews, Fife

2 M.Sc in non-numerical applications of computer science

3 A good first degree in any Faculty

4 1 calendar year

5 Suitable for graduates of any Faculty with no previous knowledge of computer science. Degree awarded on content of dissertation

6 IBM 1620 Model 2, with discs, printer and digital plotter

7 Present 4, future 15

8

First term	
Languages, introduction of systems, algorithms, non-numerical applications	50
Second term	
List processing, compilers, definition of a language	40
Third term	
Finite state machines, more non-numerical applications	20
Fourth term	
No formal lectures.	

ACKNOWLEDGEMENT

The working party would like to thank all the departments and institutions listed in the survey for their patient help in its compilation.

MEMBERS OF WP7

S. Kirkby (Chairman) *College of Aeronautics*
R. A. Brooker *University of Manchester*
R. A. Buckingham *Institute of Computer Science*
E. C. Clear Hill *Enfield College of Technology*
G. B. Cook *University of Leeds*
B. Girling *The City University*
H. L. W. Jackson *Staffordshire College of Technology*
E. B. James *Imperial College*
I. M. Khabaza *Queen Mary College*
J. C. P. Miller *University of Cambridge*
P. G. Raymont *Leicester College of Technology*
R. W. Sharp *Hatfield College of Technology*
J. M. Watts *University of Liverpool*

The Working Party would like to acknowledge the debt it owes to Mr I. M. Khabaza, under whose chairmanship this and the following article were produced.

Detailed suggestions for courses are given which it is hoped will help and stimulate the establishment of computer science education at universities and colleges. Some of these courses would be suitable for undergraduates, others for postgraduates or both.

Specimen courses in computer science

Within the wide range of standards from ancillary first year undergraduate to specialized postgraduate given here, two distinct approaches will be discerned reflecting the 'binary' trend in higher education in the United Kingdom: universities which tend to be theoretical, and colleges of technology which tend to be practical.

Introduction to Computer Science

Summary: Concept of a computer; simple programming; logic operations; flowcharts; mathematical and systems flowcharts. Operating systems; program testing. Instruction modification; storage in arrays; economic program strategy; documentation. Non-numeric problems; character codes; sorting processes. Peripherals; input and output; validity of data. Subroutines; procedures; further program testing; operating systems for production work. Description of a problem-oriented language (*e.g.* ALGOL, FORTRAN, PL 1), emphasizing local restrictions and its basic inefficiencies.

Time required and pre-requisites: The course should consist of 20 to 40 hours of which at least 50 per cent. should be spent in 2–3 hour practical lessons. For the first half of the course, at least, the practical work should be geared to an immediate turnround program testing system. Initially, students should prepare their own data and programs. The course is suitable for all types of student and if it is to form an integrated part of a course of studies it should be started during the first term and initially with a commitment of 6 hours per week.

Detailed syllabus: Introduction to a programming language and the computer. The two facets should be integrated by allowing the student to write simple programs which demonstrate: basic input and output facilities; arithmetical operations; storage and program control. The language used should be a simple teaching language and *not* languages like FORTRAN and ALGOL. If it is necessary to use such languages then a suitable *simple* subset should be used which should have a separate compiler with its own diagnostics and testing facilities. The language should be introduced slowly and initially firm restrictions adopted on input and output.

Use of flowcharts. Description of mathematical and systems problems with flowcharts. Use of British Standard symbols. A description of the *local* operating system for running test programs. The data flow through the *local* computer system. Extension of language to incorporate instruction modification. Use of arrays for data storage. Considerations of economy of storage and program instruction contrasted with programming effort. Full program documentation. This should be along the following lines:

1 Title
2 Index
3 Description of program
4 Method of use – detailed specification of data preparation, operating instructions, results format and error procedures
5 Mathematical and procedural analysis
6 Flowcharts
7 Program
8 Tests – detailed justification for tests performed and the data used with subsequent results

Non-numerical problems; descriptions of codes, binary numbers and character operations; logic operations; sorting methods. Description and use of peripherals, *e.g.* magnetic tapes, magnetic discs, graph plotters, etc, if they can be assessed by your teaching language. Design of output and input formats; consideration of data errors; validity checks. Use of subroutines, procedures and more sophisticated program testing facilities. Description of operating procedures in use *locally* for the running of production programs.

Description of a problem-oriented language (*e.g.* ALGOL, FORTRAN, PL1), emphasizing local restrictions and its basic inefficiencies.

References:

British Standards 3527. *Vocabulary*. 1962
British Standards 4058. *Flowcharts*. 1966
Gruenberger F. and Jaffray G. *Problems for Computer Solution*. Wiley, 1965
Marchant and Pegg. *Digital Computers – A Practical Approach*. Blackie, 1967
Nicol. *Elementary Programming and Algol*. McGraw-Hill, 1965

Basic Computer Programming

Summary: Use of language like FORTRAN or ALGOL. Operating systems. Program documentation. Program testing. Use of peripherals. Use of language like COBOL. Operating systems. Documentation. Program testing. Data validity. Use of peripherals. Data structure, codes, standards. Data processing. Use of an assembly language. Operating systems. Program testing. Construction of subroutines and macros. Use in indirect addressing and relocatable programs. Operation of peripherals. The techniques used in the construction of assemblers, compilers and interpreters.

Time required and pre-requisites: The course should consist of about 200 hours of which at least 70 per cent. should be spent in 2–3 hour practical sessions. When a new language is being studied the practical classes should involve immediate turn-round of test programs and facilities for the students to prepare their own test programs and corrections. One assumes that the students have all had an introductory course which has familiarized them with the basic features of the computer and computer programming. The course should be taken over a single academic year.

Detailed syllabus: The students will learn in detail three programming languages in succession. First they will learn and use a language like ALGOL or FORTRAN, then a language like COBOL: and finally an assembly language (possibly only a subset of the *local* computer's assembly language). During each study they will learn how to use the standard *official* descriptions of the language. They will operate programs through the *local* computer and become familiar with its operational system. All programs they write will be fully documented. This will involve a layout along the following lines:

1 Title and description of the program
2 Index
3 Method of use
  a) Data preparation (including validity checks)
  b) Operating instructions
  c) Description of output format
  d) Error procedures and restart techniques
4 Mathematical analysis
5 Procedural analysis
6 Flowcharts (overall and detailed where necessary)
7 Program
8 Tests (a detailed description of the tests with a justification for each one together with its data and results).

They will make full use of all test facilities and become familiar with diagnostic routines and systems documentation. Emphasis should be placed on the use of flowcharts in problem solving, economic use of the computer from various aspects such as storage, time and programming effort. Use and appreciation of all peripherals and storage media associated with the *local* computer.

Data structure; words, records, files, arrays, lists, trees; character codes; translation processes; validity checking; indirect addressing, symbolic addressing; methods of internal sorting, *e.g.* radix sorting, selection sorting, exchange sorting, sorting by insertion; methods of external sorting, *e.g.* merge sorting, sorting by insertion, polyphase and cascade sorting; methods of searching and updating. Use of subroutines, procedures and macros; relocatable and segmented programs; techniques and design characteristics of assemblers, interpreters and compilers.

References:

Arden B. W. *Introduction to Digital Computing.* Addison-Wesley, 1963
Galler B. A. *The Language of Computers.* McGraw-Hill, 1962
Iverson K. E. *A Programming Language.* Wiley, 1962
Randell B. and Russell L. J. *ALGOL 60 Implementation.* Academic Press, 1964
Wegner P. *Introduction to Symbolic Programming.* Academic Press, 1964

Computer Organization and System Programming

Summary: This is an introduction of about 40 lectures to operating systems, loading and assembly, monitoring, time-sharing, various methods of organizing input and output, store management.

Pre-requisites: Basic computer science course.

Detailed syllabus: Organization of computer hardware: properties of storage systems; flow of information through the system; methods of peripheral control; properties of peripheral devices.

Organization of computer software: translation techniques; stacks; Polish notation. Methods of optimization: storage allocation; block structure; routines; functions; compilers; interpreters, syntax directed compilers; generators; application expressions. Intermediate languages. Bootstrapping techniques.

Time sharing in a single program: autonomous input/output operations by basic instructions; user-controlled interrupts; overlapped transfers to storage media; program branching. Running under the control of a monitor system: use of clock and instruction counter interrupts; overlapping of input, output and computing of different programs; provision of debugging facilities. Multiprogramming systems: sharing the central processor system between several programs; scheduling methods for program switching; space allocation; job queueing and prescheduling; types of backing stores; scheduling of transfers to backing stores; interrupt systems.

Hardware test programming: fault finding on simple machines; on-line testing in multiprogramming systems; automatic recovery after a fault condition. Software test programming: interface with hardware; memory dumps; tracing alteration of location values and sequence control; trapping logical faults in programs.

On-line access systems: direct control from machine console; multiple direct access systems; programming aspects of direct access; filing systems. Multicomputer systems.

References:

Operating system manuals of various computers
Corbato F. J. *et al. The MULTICS system.* Fall Joint Computer Conference, 1965, pages 185–247
Fisher F. and Swindle G. *Computer Programming Systems.* Holt, Rinehart and Winston, 1964
Heistand R. E. *An executive system.* Communications ACM, 1964
The ATLAS I supervisor, operating system and scheduling system. ICT, 1966
Wegner P. *Introduction to System Programming.* Academic Press, 1964
Bucholz W. *Planning a Computer System.* McGraw-Hill, 1962

Information Structures and Processing

Summary: The representation, interrogation and transformation of information structures in a computer.

Time required: 20–40 lectures.

Pre-requisites: Course 1: Introduction to computer science.
Course 2: Basic computer programming.

Detailed syllabus: Information storage in the machine: immediate access stores; drums; discs; magnetic tape and cards.

Linear information structures: number scales; fixed and floating point representation; words and arrays. Non-numeric storage; packing and unpacking. Fixed length and variable length records.

Processing of linear information: the Iverson notation; editing; sorting; merging. Machine considerations: linear and parallel access; error control.

Complex information structures: lists, trees, list structures and their storage in the machine.

Processing of complex information: the list-processing languages LISP, IPLV, SLIP, with a special study and practical use of at least one language. The string-manipulating languages COMIT, SNOBOL, TRAC with special study and practical use of at least one language.

Transformations of information structures and deductive processes: the storage and retrieval of library materials, algebraic manipulation; formal differentiation and integration. Introduction to heuristic programming; game playing and theorem proving.

References:

Iverson K. E. *A Programming Language.* John Wiley & Sons, 1962

Bucholz W. *File organization and addressing.* IBM systems journal, Vol. 2, June 1963, pages 86–111

McCarthy J. *et al. LISP 1·5 Programmers Manual.* MIT Press, 1962

Newell A. (Ed). *Information Processing Language – V Manual.* Prentice-Hall, 1964

Weizenbaum J. *Symmetric List Processor.* Communications ACM, 6, 9, Sept. 1963

Yngve V. H. (Ed). *COMIT Programmers reference manual (COMIT I).* MIT Press, 1963

An Introduction to COMIT Programming. MIT Press, 1963

Farber, D. J., Griswold R. E. and Polonsky I. P. *The SNOBOL3 Programming Language.* Bell System Technical Journal, XIV, 6, July-August 1966, pages 895–944

Mooers C. N. *TRAC, a procedure-describing language for the reactive typewriter.* Communications ACM, 9, 3, March 1966, pages 215–219

Bobrow D. G. and Raphael B. *A comparison of list-processing computer languages.* Communications ACM, 7, 4, April 1964, pages 231–240

Brooks F. P. and Iverson K. E. *Automatic Data Processing.* John Wiley & Sons, 1963

Borko H. (Ed). *Computer applications in the behavioural sciences.* Prentice-Hall, 1962

Slagle J. R. *Experiments with a deductive question answering program.* Communications ACM, 8, 12, Dec. 1965

Bobrow D. J. *A question-answering system for high school algebra word problems.* Fall Joint Computer Conference, 1964

Green B. F. *et al. Baseball: an automatic question answerer.* Western Joint Computer Conference, 1961

Programming Languages and Compilers

Summary: Theory of grammars. Language structure and content. Structure of translator (lexicographical analysis and editing, syntactical analysis, processing the results of analysis). Structure of object program. Special purpose languages.

Time required: Up to 80 lectures. Figures in brackets indicate relative importance of material.

Pre-requisites: An introductory course on computers.

Detailed syllabus: Theory of grammars: Chomsky type 0, 1, 2 (simple phrase structure) and 3 (finite state grammars). Syntax specification by BNF and variants. Grammars admitting of special parsing algorithms such as precedence analysis. (4 lectures.)

Language structure and content: lexicography and syntax. Routine, blocks, compound statements. Scope of declarations. Types of formal parameters. Conditions. Expressions. Primitive data types. Compound data types (structures). Parallelism. Further definitional facilities; macros. I/O and peripheral operations (provided by library routines?). Supervisor operations. Fault trapping. (22 lectures.)

Structure of translator:

1 Lexicographical analysis and editing of the elementary syntactic units, *e.g.* identifiers, constants, etc.

2 Methods of syntactic analysis. Syntax controlled analysis: methods based on operator precedence or other special cases of what Floyd calls bounded context syntactic analysis. Syntax directed analysis: bottom-up and top-down methods. Recasting the grammatical definitions to simplify these methods.

3 Processing the results of analysis. This may be some form of parenthesis-free notation which can be interpreted directly or converted to machine code. Formal semantic systems, *e.g.* Compiler-Compiler, FSL, COGENT. Auxiliary tables: identifiers and property lists. Methods combining (2) and (3), *e.g.* Grau and Floyd-Evans. (17 lectures.)

4 Treatment of macros.

Structure of object program: storage allocation (the stack, fixed storage, garbage collection). Block and routine administration. Transfers of control. Elementary arithmetic. Cycles and subscripting. Arrays. Optimization. Representation of compound data structures. The system routines for I/O, peripheral, and other operations. Fault trapping. Parallelism. (15 lectures.)

Special purpose languages: simulation (*e.g.* SIMULA ALGOL, SOL, CSL). Survey analysis (*e.g.* MVC), symbolic and algebraic manipulation (*e.g.* COMIT, AMBIT, FORMAC, LISP). Essential features of data processing languages. Information retrieval languages. On-line languages. (22 lectures.)

References:

Cheatham T. E., Jr. *The Theory and Construction of Compilers.* (Private publication.) Computer Associates, 1966

Bolliet L. *Compiler Writing Techniques.* Institutes de Mathematiques Appliquees, 1966

Notes for NATO Summer School on Programming Languages, Grenoble

Goodman R. (Ed). *Annual Review in Automatic Programming* Vols. 1, 2, 3, 4. Pergamon Press, 1960–1965

Several interesting papers will be found in these volumes.

Randall B. and Russell L. J. *ALGOL 60 Implementation.* Academic Press, 1964

Baumann T., Feliciano M., Bauer F. L. and Samelson K. *Introduction to ALGOL.* Prentice-Hall, 1964

Wegner P. *Introduction to System Programming.* Academic Press, 1964

Teichroew D. and Lubin J. F. *Computer Simulation – Discussion of the Technique and Comparison of Languages.* Communications ACM, 9, 10, October 1966, pages 723–741

Computer Languages for Symbolic and Algebraic Manipulation. Special issue of Communications ACM, 9, 7, July 1966

Colin A. J. T. *Multiple Variate Counter.* University of London, Institute of Computer Science, 1964

Hoare C. A. R. *Record Handling.* Lectures for NATO Summer School, Grenoble, 1966

Griffiths T. C. and Petrick S. R. *On the Relative Efficiencies of Context-free Grammar Recognizers.* Communications ACM, 8, 5, May 1965

Floyd R. W. *The Syntax of Programming Languages – A Survey.* IEEE Transactions on Electronic Computers, EC-13, 4, August 1964, pages 346–353

In addition to the above there are numerous books describing the standard international languages.

Information Theory

Summary: Information must be sent to a receiver distant in space or time. There are three stages: encoding; transmission or storage; and recovery and/or decoding.

Messages may be *continuous* or *digital.* We may divide the theory into three parts: *Communication Theory.* The statistical treatment of signal and noise. Determination of capacity and efficiency of channels. Shannon's theorem. This theory of continuous messages is not very relevant to digital computer work. *Mathematical theory of codes. Switching theory.*

Time required: About 40 lectures.

Pre-requisites: Some knowledge of abstract alg bra.

Detailed syllabus:

1 Theory of codes: information channels. Types for theoretical discussion. *a*) Binary symmetric channel, *b*) binary erasure channel. Error detection and error correction. Maximum likelihood decoding.

Binary shift registers. These are useful for the production of sequences of binary digits for *messages* or for *pseudo-random sequences*, and also for decoding.

Special codes. Block codes. Hamming distance. Linear group codes. Slepian's theory. Group structure. Cyclic codes. Use of binary shift registers for coding and decoding. Meggitt's examples.

Binary sequences. Theory. Generating functions. Sequences by division. Selmer's fundamental identities. Recurrence relations; classical method. Periods and properties. Pseudo-random sequences. Golomb's results.

Further special codes. Useful properties of error-correcting codes. Hamming codes. Reed-Muller codes. Iterated codes. Shortened cyclic codes, etc.

2 Switching theory: idea of a sequential machine as a processor of time or space sequences of digits. Requirements for combination, sequencing and storing of digit patterns.

Combinational logic and measures of complexity for combinational nets; *e.g.* gate count, arrowhead count. Fan power and logic depth considerations. Methods for describing, synthesizing and simplifying combinational functions. Hazards.

Mealy machines, Moore machines, etc. Number of states in a sequential net. Procession of states. State diagrams and transition tables. Production of the reduced machine (a canonical form). Assignment of states to flipflops (not a completely solved problem, but contributions to state assignment are numerous and useful).

Accommodation of non-ideal signals. Synchronous, asynchronous and speed-independent networks. Huffman's theory of asynchronous networks. Muller's theory of speed independence. Post algebra and Eichelburger's ternary description logic. Race conditions.

References:

The general background was extracted from Golomb 1964, with help from Peterson 1961; the number theory of binary sequences was taken mainly from Selmer 1966, but with reference to Peterson 1961, and MacWilliams 1965. Details of special error-correcting codes are given very fully in Peterson 1961; Slepian 1956 gives a readable account of group codes; Meggitt 1960–61 was found a useful paper for providing detailed examples, while MacWilliams 1965 uses high-power mathematics in a practical way to determine sequences with specified desired properties.

For switching theory McCluskey 1965 is recommended as a text-book, with other references for special topics

Theory of Codes

Golomb S. W. *et al. Digital Communications with Space Applications.* Prentice-Hall, 1964

MacWilliams J. *The structure and properties of binary cyclic alphabets.* Bell System Technical Journal, 44, pages 303–332, 1965

Meggitt J. E. *Error-Correcting Codes and their Implementation for Data Transmission Systems.* A report stamped IBM British Laboratories. About 1960–1961

Peterson W. W. *Error-Correcting Codes.* MIT Press, 1961

Selmer E. S. *Linear Recurrence Relations over Finite Fields.* Dept. of Mathematics, University of Bergen, 1966

Slepian D. *A class of binary signalling alphabets.* Bell System Technical Journal, 35, pages 203–234, 1956

Switching Theory

Curtis H. A. *The Design of Switching Circuits.* Van Nostrand, 1962

Eichelburger E. B. *Hazard detection in combination and sequential switching circuits.* IBM J. Res. and Dev, 9, pages 90–99, 1965

Huffman D. A. *The synthesis of sequential switching circuits.* J. Franklin Inst., 257, pages 151–190, 275–303, 1954

McCluskey E. J. *Introduction to the Theory of Switching Circuits.* McGraw-Hill, 1965

Mealy G. H. *A method for synthesizing sequential circuits.* Bell System Technical Journal, 34, pages 1045–1079, 1955

Moore E. F. *Gedanken experiments on sequential machines. Automata studies.* Annals of Math, 34, pages 129–153, 1956

Phister M. *Logical Design of Digital Computers.* Wiley, 1958

Numerical Methods

Summary: This is a course of about 40 lectures on numerical methods for solving linear and non-linear equations, matrix computations, ordinary and partial differential equations, iterative methods. Although students' attention would be drawn to the various limitations of the methods, the emphasis would be on the algorithms rather than mathematical error analysis.

Pre-requisites: Introduction to computer science.

Detailed syllabus: Linear algebra; matrix algebra; direct and iterative methods for the solution of sets of linear algebraic equations and the inversion of matrices; elementary theory of eigen values and eigen vectors and direct and iterative methods for their determination.

Methods of approximation: polynomial approximation by difference and Lagrangian methods, including applications to interpolation, numerical differentiation and numerical integration; orthogonal polynomials; Gaussian quadrature; method of least squares; harmonic analysis; rational approximations.

Solution of non-linear equations: determination of roots of non-linear equations by iterative methods; successive substitution, rule of false position, Newton-Raphson, modified Newton, Bairstow's and Muller's methods. Aitken's δ^2 process. Analysis of convergence and comparison of iterative methods.

Ordinary differential equations: analytic methods. Taylor's series and Picard's method. Single and multistep discrete variable methods (Runge-Kutta, predictor-corrector, etc); the difference correction method; sets of simultaneous equations and equations of higher order; special methods for second order equations; stability. Two point boundary-value problems; characteristic values and functions; approximate solution using difference methods; difference correction; use of linear compounds in approximating to solutions.

Introduction to partial differential equations: quasi-linear equations of the first order; classification of quasi-linear equations of the second order and discussion of the appropriate ancillary boundary conditions; characteristics; pair of simultaneous first order equations. Finite difference approximations; variational methods for the solution of elliptic equation; step-by-step forward integration of parabolic equations and the problem of stability; elementary applications of the method of characteristics to the solution of hyperbolic equations.

Elementary ideas of error and precision.

References:

Bull G. *Computational Methods and ALGOL.* Harrap, 1966

Noble B. *Numerical Methods.* Vols. 1 and 2. Oliver & Boyd, 1964

Hamming R. W. *Numerical Methods for Scientists and Engineers.* McGraw-Hill, 1962

Hildebrand F. B. *Introduction to Numerical Analysis.* McGraw-Hill, 1956

National Physical Laboratory. *Modern Computing Methods.* HMSO, 1961

Ralston A. *A first course in Numerical Analysis.* McGraw-Hill, 1965

Fox L. *An Introduction to Numerical Linear Algebra.* Oxford, 1965

Smith G. D. *Numerical Solution of Partial Differential Equations.* Oxford, 1965

Numerical Analysis

Summary: This is a course of about 40 lectures on the more advanced parts of numerical analysis requiring more mathematical background, and emphasizing error analysis and convergence problems.

Pre-requisites: Elementary numerical methods course.

Detailed syllabus: Error analysis in linear algebra: norms of vectors and matrices. Measures of ill-conditioning of matrices, eigen vectors and eigen values. Error analysis of elimination methods and of methods for computing eigen values.

Ordinary differential equations: error propagation for discrete variable methods; stability and convergence; statistical theory of the propagation of rounding errors.

Partial differential equations: more advanced treatment of elliptic equations including methods in current use (SOR, Chebyshev, ADI, Peaceman-Rachford, etc). Problems of convergence and stability for parabolic equations. More advanced treatment of hyperbolic equations; characteristic co-ordinates; elaboration of method of characteristics. Introduction to treatment of shock waves. Eigen value problems; treatment of singularities.

Integral equations: numerical methods for solving integral equations.

References:

Forsythe G. E. and Wasow W. R. *Finite Difference Methods for Partial Differential Equations*. Wiley, 1960

Fox L. *Numerical Solution of Ordinary and Partial Differential Equations*. Pergamon Press, 1962

Todd J. *Survey of Numerical Analysis*. McGraw-Hill, 1962

Varga R. *Matrix Iterative Analysis*. Prentice-Hall, 1962

Wendroff B. *Theoretical Numerical Analysis*. Academic Press, 1966

Wilkinson J. H. *Rounding Errors in Algebraic Processes*. HMSO, 1963

The Algebraic Eigen value Problem, Oxford, 1965

Walsh Joan *Numerical Analysis*. Academic Press, 1966

Statistical Data Processing and Analysis

Summary: Computational aspects of the analysis of data; data screening and checking; survey analysis; multivariate analysis; analysis of variance; regression analysis; time series analysis.

Time required: 40 hours plus 20 hours practical work.

Pre-requisites: An introductory course on statistics of about 40 hours, including practical work; this would have covered statistical inference, experimental designs, descriptive statistics, hypothesis testing, estimation, regression and correlation.

Detailed syllabus: Survey analysis: sampling techniques, data collection, coding and checking. Descriptive techniques: graphic, tabular and other outputs. Multivariate analysis: computational aspects, formation of covariance and correlation matrices, principal component analysis, factor analysis and rotation, discriminant analysis and distance techniques for cluster analysis. Analysis of variance: up to three factor simple designs, Latin squares, missing observations dealt with by use of design matrix. Response surface analysis. Regression analysis: multiple linear regression with inclusion or exclusion of variables, step-wise regression, use of dummy classification variables, non-linear regression. Time series analysis: spectral analysis, moving averages, smoothing, trend estimation and seasonal adjustment, other cyclical variations, index numbers and forecasting.

References:

Manufacturers' literature giving program specifications, and the following:

Yates F. *Sampling Methods for Censuses and Surveys*. Griffin & Co, 1959

Lawley D. N. and Maxwell A. E. *Factor Analysis as a Statistical Method*. Butterworths, 1963

Kendall M. G. *A Course in Multivariate Analysis*. Griffin & Co, 1957

Davies O. L. (Ed). *The Design and Analysis of Industrial Experiments*. Oliver & Boyd, 1956

Davies O. L. *Statistical Methods in Research and Production*. Oliver & Boyd, 1957

Smillie K. W. *An Introduction to Regression and Correlation*. Academic Press, 1966

Kendall M. G. and Stuart A. *The Advanced Theory of Statistics*. Vols. I, II and III. Griffin & Co, 1959, 1961, 1966

Data Processing

Summary: File processing, serial and random-access. COBOL programming. Sorting. Sort and report generators. Systems flowcharts. Decision tables. Simple data processing applications. Data processing equipment. Data preparation. Accounting machines. Electronic calculators.

Time required: 60 lectures with supporting practical work.

Pre-requisites: An introductory course on computers.

Detailed syllabus: Files, blocks and records. File updating. Master and transaction files. Systems flowcharts. COBOL – an introduction for punched card files. Punched card data processing. Keypunches and verifiers. The sorter, collator and tabulator. Plugboard wiring. Paper tape data processing. Accounting machines and electronic calculators. Magnetic tape operations in COBOL. Magnetic tape file processing. Sorting and collating. Random-access file processing. Sort and report generators. Applications to payroll, sales analysis, accounting and invoicing, inventory control. Systems design and documentation. Decision tables. Operation of on- and off-line equipment. Tape drives, discs and drums. Teleprinters and line printers. Console typewriters. Card and paper tape readers and punches. Optical and magnetic ink character readers. Graph plotters and graphic display devices. Data links and communications equipment.

References:

In addition to the books listed below the various manufacturers' publications should be consulted.

Awad E. M. *Business Data Processing*. Prentice-Hall, 1965
A very useful introduction

Brooks F. P. and Iverson K. E. *Automatic Data Processing*. Wiley, 1963
An excellent survey

Burton A. J. and Mills R. G. *Electronic Computers and their Business Applications*. Benn, 1960
Contains much useful information on applications although published some time ago

Gregory R. H. and van Horn R. L. *Business Data Processing and Programming*. Wadsworth, 1963
Another excellent survey

McCracken D. D. *A Guide to COBOL Programming*. Wiley, 1963
Deals with IBM 7090 COBOL but has many useful examples and exercises

McCracken D. D. *An Introduction to IBM 1401 Programming*. Wiley, 1962
Contains much useful general information besides the specific 1401 material

Management Data Processing

Summary: Business systems; case studies. The integrated system concept. Systems analysis: feasibility studies, systems design, documentation and proving. Management science: planning and forecasting, decision making, simulation, heuristic programming.

Time required: 30–40 lectures and supporting practical work.

Pre-requisites: Courses on file processing and COBOL programming and on data processing equipment; also a parallel course on statistics and operational research.

Detailed syllabus: Business systems: systems flowcharts and decision tables. Software for commercial programming: sort and report generators, input/output control systems. Case studies, *e.g.* payroll, sales analysis, invoicing, inventory control, production control (to include systems documentation and COBOL programs). The concept of an integrated system. Communications and real-time systems.

Systems analysis: feasibility studies. Fact finding. Systems design and documentation. Timing considerations; choice of equipment. Programming and conversion. Validation and error procedures. Systems proving, operation and maintenance. Use of applications packages. Special problems of real-time systems.

Management Science Applications: the collection of information and the presentation of results. Planning and forecasting: statistical forecasting techniques. Linear programming; the implementation of the simplex method. Critical path methods and PERT. Decision making and the use of models. Business games. Simulation programming: the use of a sim[illegible]nguage, *e.g.* CSL. Heuristic programming: [illegible]tegies and recursion; applications (*e.g.* the line[illegible]oblem).

References:

Many of the books below have bibliographies giving further references. In addition, the computer manufacturers' publications may be consulted.

Brandon. *Management Standards for Data Processing.* V. Nostrand, 1963
Very useful on systems analysis

Burton and Mills. *Electronic Computers and their Business Applications.* Benn, 1960
Gives details of some applications

Feigenbaum and Feldman. *Computers and Thought.* McGraw-Hill, 1963
Contains papers on heuristic programming

Fisher and Swindle. *Computer Programming Systems.* Holt, Rinehart & Winston, 1964
Useful on commercial software

Laden and Gildersleeve. *System Design for Computer Application.* Wiley, 1963
Excellent coverage of systems analysis

McCracken. *COBOL Programming.* Wiley, 1963
Contains useful case studies

McMillan and Gonzalez. *Systems Analysis.* Irwin, 1966
Useful on models and simulation

Operations Research

Summary: This is an introductory course on basic methods of Operations Research, mathematical programming, network analysis, critical path planning and scheduling, queueing methods and simulation.

Time required: 30–40 lecture hours or about one-sixth of a year's work load for a university student.

Pre-requisite: An introductory statistics course.

Detailed syllabus: (*N.b.* The emphasis given to the topics will depend on the interests of the students and on their mathematical and statistical knowledge. There is a large overlap with the course on mathematical optimization methods.)

The scope and limitations of operations research. Place of mathematics in OR. Nature of available information and methods of obtaining it. (2 lectures.)

Introduction to graph theory; shortest path algorithm, critical path methods, float and slack. PERT/time, PERT/cost. Network flows; the maximum flow – minimum cost theorem, Ford-Fulkerson algorithm. (8 lectures.)

Linear programming; examples, mathematical formulation, the simplex algorithm with geometric explanation, extensions of the linear programming technique, the transportation problem. Dynamic programming with examples of its use. (9 lectures.)

Queueing theory; the Poisson distribution, branching processes, simple birth and death processes. (4 lectures.)

Inventory control; simple examples of the calculation of re-order quantity in the single and multi-product case, when the demand is fixed or probabilistic. (4 lectures.)

Renewal theory; replacement with catastrophic failures and with increasing maintenance costs. (4 lectures.)

Simulation; basic ideas, uses and limitations. Planning simulation experiments. Statistical techniques, generation of pseudo-random numbers. Variance reduction techniques. Examples of the use of simulation. Computer simulation languages, *e.g.* SIMULA ALGOL, SOL, CSL, SIMSCRIPT or DYNAMO. (9 lectures.)

References:

Sasieni M., Yaspan A. and Freidman L. *Operations Research: Methods and Problems.* Wiley, 1959
A good text for this course

Saaty T. L. *Mathematical Methods of Operations Research.* McGraw-Hill, 1959
Very advanced

Kaufmann A. *Methods and Models of Operations Research.* Prentice-Hall, 1963

Vazsonyi A. *Scientific Programming in Business and Industry.* Wiley, 1958

Churchman C. and Verhulst M. *Management Sciences, Models and Techniques.* 2 Vols. Pergamon, 1960

Kantorovitch L. V. *The Best Use of Economic Resources.* Pergamon, 1965

Gass S. I. *Linear Programming.* McGraw-Hill, 1964

Nemhauser G. L. *Introduction to Dynamic Programming.* Wiley, 1966

Llewellyn R. *Linear Programming.* 1964

Dantzig G. *Linear Programming and Extensions.* Princeton UP, 1963

Cox D. R. *Renewal Theory.* Methuen, 1962

Graves R. L. and Wolfe P. (Ed). *Recent Advances in Mathematical Programming.* McGraw-Hill, 1963
Very advanced

Arrow K. J., Karlin S. and Scarf H. *Studies in the Mathematical Theory of Inventory and Production.* Stanford UP, 1958

Wagner H. M. *Statistical Management of Inventory Systems.* Wiley, 1962

Hadley G. F. and Whitin T. McL. *Analysis of Inventory Systems.* Prentice-Hall, 1963

Elmaghraby S. E. *The design of Production Systems.* Reinhold, 1966

Karlin S. *Mathematical Methods and Theory in Games, Programming and Economics.* Adison-Wesley, 1959

Kibbee J. M., Craft C. L. and Nanus B. *Management Games.* Reinhold, 1961

Network Theory

Lockyer K. G. *Introduction to Critical Path Analysis.* 1965

Moder J. J. and Phillips C. R. *Project Management with CPM and PERT.* Chapman & Hall, 1964

Battersby A. *Network Analysis for Planning and Scheduling.* McMillan, 1964

Busacker R. G. and Saaty T. L. *Finite Graphs and Networks.* McGraw-Hill, 1965

Berge C. *The Theory of Graphs.* Methuen, 1962
Difficult

Ore O. *Graphs and Their Uses.* Random House, 1963

Beckenback E. F. (Ed). *Applied Combinatorial Mathematics.* Wiley, 1964

Ford L. R. and Fulkerson D. R. *Flows in Networks.* Princeton, 1962

Queueing Theory and Simulation

Khinchin. *Mathematical Methods in the Theory of Queueing.* Griffin, 1960

Riordan. *Stochastic Service Systems.* 1962

Saaty T. L. *Elements of Queueing Theory.* 1961

Lee A. M. *Applied Queueing Theory.* 1966

Cox D. R. and Smith W. L. *Queues.* Methuen, 1961
Very good

Hall and Dobell. *SIAM Review IV.* 1962, pages 230–254
A good review

Tocher K. D. *The Art of Simulation.* EUP, 1963
Good

Shreider I. A. (Ed). *The Monte Carlo Method: The Method of Statistical Trials.* Permagon, 1966

Naylor T. H., Balintfy J. L., Burdick D. S. and Kong Chu. *Computer Simulation Techniques.* Wiley, 1966
Good

Chorafas D. N. *Systems and Simulation.* Academic Press, 1965

Hammersley J. M. and Handscomb D. C. *Monte Carlo Methods.* Methuen, 1964

Mathematical Optimization Methods

Summary: Introduction; general methods for unconstrained and constrained problems; linear and non-linear programming; dynamic programming.

Time required and pre-requisites: This could be a course of about 30–40 lecture hours, or about one-sixth of the year's work-load for final year university mathematics students. For other classes the syllabus and time allowed must be modified accordingly.

Detailed syllabus: The aim throughout should be to integrate theory with the performance of the methods described in practice and with their realization as computing algorithms. Examples should be given of the success and failure of the methods considered.

Description of the optimization problem. Local and global extrema. Minimization of a function of a single variable. Fibonacci search, interpolating polynomial methods. Unconstrained optimization, classical methods, quadratic functions, generalized Newton-Raphson method. Methods using first derivative only. Variable metric methods of Davidon and others. Methods using function values only. Constrained minima. Reduction to unconstrained form, Lagrange multipliers, projection methods. (11 lectures.)

Linear programming: the linear programming problem; canonical forms; vector spaces and convex sets; the simplex method; artificial variables; degeneracy; cycling; perturbation techniques; duality problems; the revised simplex method; product form of the inverse; parametric linear programming; sensitivity analysis; the dual simplex method; primaldual algorithm; *decomposition algorithm*; the transportation problem; stepping-stone method; *integer programming*; cutting plane; primal branch and bound; and partial enumeration methods. (22 lectures.)

Quadratic programming; convex functions; Kuhn-Tucker theory. (2 lectures.)

Dynamic programming: principle of optimality; functional equations and computational procedure; one-dimensional allocation processes; multi-dimensional allocation processes; Lagrange multipliers. (5 lectures.)

(*N.b.* – The italicized topics are of an advanced nature and can be omitted if desired.)

References:

Walsh J. (Ed). *Numerical Analysis: An Introduction.* Academic Press, 1966

Chapter 8 by M. J. D. Powell and its references. A good general survey.

Spang H. A. III. *A Review of Minimization Techniques for non-linear functions.* SIAM Review 4, 1962, page 343

Balakrishnan and Neustadt (Eds). *Computing Methods in Optimization Problems.* Academic Press, 1964

Leitman G. *Optimization Techniques with Applications to Aerospace Systems.* 1962

Lavi and Vogl (Ed). *Symposium on Recent Advances in Optimization Techniques.* Pitsburg, 1965

Pontriagin L. S. *et al. The Mathematical Theory of Optimal Processes.* Interscience (John Wiley), 1962

Zaguskin. *Handbook of Numerical Methods for the Solution of Algebraic and Transcendental Equations.* 1961

Linear Programming

Llewellyn R. *Linear Programming.* 1964

A good basic text

Gass S. I. *Linear Programming.* McGraw-Hill, 1964

An excellent text on the *theory* of LP.

Dantzig G. *Linear Programming and Extensions.* Princeton UP, 1963

Hadley G. *Linear Programming.* Adison-Wesley, 1962

Hadley G. *Non-Linear and Dynamic Programming.* Adison-Wesley, 1964

Dynamic Programming

Shuchman A. M. (Ed). *Scientific Decision Making in Business.* Holt, Rinehart and Winstone, 1963

Contains an excellent chapter on dynamic programming

Nemhauser G. L. *Introduction to Dynamic Programming.* Wiley, 1966. A basic text

Jacobs O. L. R. *An Introduction to Dynamic Programming.* 1967

Bellman R. and Dreyfus S. *Applied Dynamic Programming.* 1962

Dreyfus S. *Dynamic Programming and the Calculus of Variations.* Academic Press, 1965

Logic Design and Switching Theory 1

Summary: Boolean algebra. Analysis and synthesis of logic networks. Combinational and sequential switching systems. Design of counters and shift register. Computer sub-systems. Methods of executing arithmetic operations.

Time required: 40 hours lectures with supporting practical work.

Pre-requisites: General mathematical background plus, preferably, some contact with electronics or electrical networks.

Detailed syllabus: Boolean algebra. Postulates, theorems, canonical forms. Function representation and minimization by algebraic, graphical and tabular methods. Logic realization. Types of gate and appropriate synthesis methods. Hazards. Contact networks, special forms. Bistable elements, logic description and application to counters and shift registers. Arithmetic operations, serial and parallel adders and subtractors. Multiplication and division methods. Sequential machines, Moore and Mealy models. Synthesis procedures for asynchronous and synchronous machines, hazards. Practical methods of realization.

References:

Chu Y. *Digital Computer Design Fundamentals.* McGraw-Hill, 1962

Good coverage of combinational logic, with computer applications. Excellent section on arithmetic methods

Coldwell S. K. *Switching Circuits and logical design.* Wiley, 1958

A classic giving thorough treatment of all aspects of contact networks but weak on electronic realizations. Many exercises

Flegg H. G. *Boolean Algebra.* Blackie, 1964

Fairly formal approach covering mainly combinational logic. Introduces Boolean matrix methods. Exercises provided

Maley G. C. and Earle J. *The Logic Design of Transistor Digital Computers.* Prentice-Hall, 1963

Concentrates on NOR and NAND network realization

Miller R. E. *Switching Theory, Vol. I.* Wiley, 1965

Excellent formal text on combinational logic with many exercises

Logic Design and Switching Theory 2

Summary: Extension and rounding out of material of Course 1.

Time required: 15 to 20 lectures with practical work.

Pre-requisites: Course 1 or equivalent.

Detailed syllabus: Special Boolean functions and their implications to engineering. Boolean functions classified according to form of network. Linear and non-linear feedback shift registers. Refinements in arithmetic sub-systems to increase computing speed. Sequential machines, minimal state theorems, incomplete specification, the state assignment problem, regular expressions. Hazards in pulsed sequential circuits. Design for reliability, redundancy, level-operated circuits.

References:

Flores J. *The Logic of Computer Arithmetic.* Prentice-Hall, 1963

A very thorough exposition of modern methods in arithmetic units

Miller R. E. *Switching Theory, Vol. II.* Wiley, 1965

Not light reading but one of the best formal treatments of sequential machines in book form

Analog and Hybrid Computing

Summary: The course covers analog and hybrid programming with some hardware being considered.

Time required: 60 lectures with supporting practical work (approximately 20 hours).

Pre-requisites: A general engineering or mathematical background involving some knowledge of elementary digital computer programming using a language such as EMA, ALGOL or FORTRAN.

Detailed syllabus: Linear elements: operational amplifiers, potentiometers, linear differential equations with constant coefficients. Control: input/output peripherals, control modes and circuits. Scaling and linear analysis: problem and machine units; time and amplitude scaling and frequency approximation; linear analysis by operator, time domain, frequency domain and transform techniques.

Non-linear elements: multipliers, resolvers, diodes and relays; function generation by explicit and implicit techniques; time varying and non-linear differential equations; transport delays.

Error analysis: dynamic and static checks; error analysis and instability; noise and performance evaluation; error reduction techniques.

Advanced analog techniques; repetitive operation; direct simulation; optimization and control system studies; adjoint techniques.

Hybrid computer systems: analogue computers with logic modules and/or with independent digital control computers; mixed computing systems involving time sharing; analog-digital and digital-analog conversion; hybrid interfaces, hybrid software requirements; subservient digital programs.

Hybrid applications: general iterative techniques, boundary value and eigen value problems; partial differential equations; control and optimization techniques.

References:

Fifer S. *Analogue Computing – 4 vols.* McGraw-Hill, 1961
This is *the* standard reference

Johnson C. L. *Analog Computer Techniques.* McGraw-Hill, 1963
Very good for servomechanisms and the simulation of electronic devices

Levins L. *Methods for solving Engineering Problems using Analog Computers.* McGraw-Hill, 1964
Engineering orientated – very sound presentation

Tomovic and Karplus. *High Speed Analog Computations.* Wiley, 1962
Error analysis – excellent

Rogers and Connelly. *Analog Computation in Engineering Design.* McGraw-Hill, 1960
Good all round text

Paul R. J. A. *Fundamental Analogue Techniques.* Blackie, 1965
Excellent for direct simulation and transfer function realization

Stewart and Atkinson. *Basic Anaglone Computer Techniques.* McGraw-Hill, 1967
A good inexpensive text

Real-time Programming

Summary: Application of computer and programming systems to real-time, on-line and multi-access working in commercial, industrial defence and transportation problems.

Time required: 30 lectures plus treatment of special topics.

Pre-requisites: A basic knowledge of computers, ancillaries, programming and data processing.

Detailed syllabus: System reliability; redundancy, failure modes, switchovers, fault recovery. Control of multiple input/output devices; priorities, overloads, queues. Random access storage; organization and integrity. Multi-programming. Supervisory, control or executive programs. Integrated systems; matching, documentation, program testing. Design problems and simulation of multi-program operations. Errors; sources, identification and correction.

Selected topics from: Aircraft, ship and submarine navigation. Weapons firing and trajectory control. Spacecraft trajectory control. Early warning radar systems. Industrial process control. Air, road and rail traffic control systems. Airline seat reservations and operations control. Production scheduling and control. Financial transactions; banking and stock exchange. Design of engineering components and assemblies. Teaching machines. Multi-access computing.

References:

In addition to the books listed below the various manufacturers' programming manuals should be consulted.

Martin J. *Programming Real-Time Computer Systems.* Prentice-Hall, 1965

Desmonde W. H. *Real-Time Data Processing Systems – Introductory Concepts.* Prentice-Hall

Martin J. *Systems Analysis for Real-Time Computers.* Prentice-Hall

Mathematical Logic

Summary: This course covers the propositional and predicate calculuses, recursive arithmetic, Turing machines and finite automata.

Time required: Approximately 40 lectures.

Pre-requisites: A general mathematical background of engineering degree standard.

Detailed syllabus: Introduction: algebra of sets, Boolean algebra. Propositional algebra: the notations of Hilbert and Lukasiewics, connectives, duality and normal forms.

Propositional calculus: the concept of a formula, deduction theorem, equiveridic formulae, consistency and completeness of propositional calculus, independence of axioms of propositional calculus.

Predicate logic: predicates, quantifiers, axioms, normal form. Restricted calculus of predicates: formulae, axioms, consistency, quantification, deduction theorem.

Recursive arithmetic and finite automata: primitive and general recursive formulae, Turing machines, computation of recursive formulae by Turing machines, recursive sets and predicates, Church's λ-notation and calculus, finite automata, state tables, synchronous sequential circuits, reduced forms.

References:

Church A. *Mathematical Logic.* Princeton, 1956
A classical text book on the subject: original material

Kneebone G. T. *Mathematical Logic and the Foundations of Mathematics.* Van Nostrand, 1963
Extremely lucid and well documented

Novikon P. S. *Elements of Mathematical Logic.* Oliver & Boyd, 1964

Gill A. *Introduction to theory of Finite State Machines.* McGraw-Hill, 1962
A standard text book on the subject.

Shannon and McCarthy. *Automata Studies.* Princeton, 1956
An early work on automata; very sound

Mendelson E. *Introduction to Mathematical Logic.* Van Nostrand, 1966

Davis M. *Computability and Unsolvability.* McGraw-Hill, 1958

ACKNOWLEDGEMENTS

The working party would like to thank many individuals and computer science departments and especially to acknowledge its debts to the ACM report on an undergraduate programme in computer science published in the *Communications of the ACM*, Volume 8 No. 9, September 1965.

professeurs

60	personnes participent à l'enseignement pour les sections d'Ingénieurs-mathématiciens :
13	Professeurs de la Faculté des Sciences
20	Maîtres-assistants et Assistants de Faculté
12	Ingénieurs du centre de recherche
12	Ingénieurs extérieurs (secteur public ou privé)
3	Professeurs de l'Enseignement Technique.

programmes

SECTION NORMALE

Première année

TECHNIQUES MATHÉMATIQUES DE LA PHYSIQUE *

Probabilités, intégration, matrices, variable complexe, analyse vectorielle, fonctions de Bessel, systèmes différentiels, équations aux dérivées partielles.

ÉLECTRICITÉ *

Lois différentielles et intégrales de l'électrostatique et de l'électromagnétisme, milieux diélectriques et magnétiques, équations de Maxwell, l'électron et les ions, radioélectricité, théorie des solides.

ANALYSE NUMÉRIQUE *

Calcul matriciel, inversion de matrices, résolution de systèmes linéaires : méthodes directes, méthodes de relaxation, interpolation, dérivation, intégration, équations différentielles.

INITIATION A LA PROGRAMMATION (Algol, CAB 500)

INTRODUCTION A L'INFORMATIQUE

ARITHMÉTIQUE APPLIQUÉE

ALGÈBRE DE BOOLE

THÉORIE DES RÉSEAUX

CALCUL OPÉRATIONNEL

ESPACES VECTORIELS NORMÉS

INTRODUCTION A L'ÉLECTRONIQUE

TECHNOLOGIE ET DESSIN

* Le programme de ces cours permet de se présenter au certificat de licence du même nom.

Deuxième année

MATHÉMATIQU[illegible]

Ensembles, structures algébrique[illegible] [illegible]aces métriques, espaces normés, séries, intégration, [illegible]ce des solutions d'équations différentielles.

LOGIQUE ET PROGRAMMATION *

Systèmes formels, calculabilité, machines de Turing, théorie des automates et langages formels.

TECHNIQUES DE LA PROGRAMMATION

Description complète du calculateur I.B.M. 7044, langage d'assemblage MAP, programmation du système IBSYS, téléprocessing.

CALCULATRICES DIGITALES

Circuits de calcul, techniques des opérateurs, des mémoires, des organes d'entrée-sortie, organisation logique des systèmes digitaux.

CALCULATRICES ANALOGIQUES

Opérateurs linéaires et non-linéaires, résolution de systèmes différentiels et aux dérivées partielles, de systèmes algébriques, simulation.

ÉLECTRONIQUE IMPULSIONNELLE ET MÉMOIRES

Transistors, circuits impulsionnels et rapides, mémoires à grande capacité, mémoires à tores, circuits logiques magnétiques.

TECHNIQUES DE CALCUL, TECHNIQUES D'ORGANISATION

RECHERCHE OPÉRATIONNELLE

TECHNOLOGIE ET DESSIN SPÉCIALISÉS

Technologie des machines à calculer, plans de mécanismes ou de circuits, projets d'organes de machines, rédaction de notices.

Troisième année

Enseignements communs aux trois options :

STATISTIQUES *

Principales lois statistiques, échantillons empiriques, tests d'hypothèse, tests paramétriques ou tests d'adéquation, théorie de l'estimation.

MÉTHODES MATHÉMATIQUES DE LA PHYSIQUE *

Intégration, distributions, convolution, espaces de Hilbert, fonctions spéciales, équations intégrales.

PROGRAMMATION – COMPILATION

THÉORIE DES LANGAGES

PRÉPARATION DU PROJET

OPTION :

CALCUL SCIENTIFIQUE

Théorie de l'approximation.
Théorie des valeurs propres.
Equations différentielles et aux dérivées partielles.
Equations de la mécanique.
Méthodes de Monte-Carlo.
Programmation linéaire.
Programmation dynamique.
Problèmes en recherche opérationnelle.
Analyse de la variance.

OPTION :

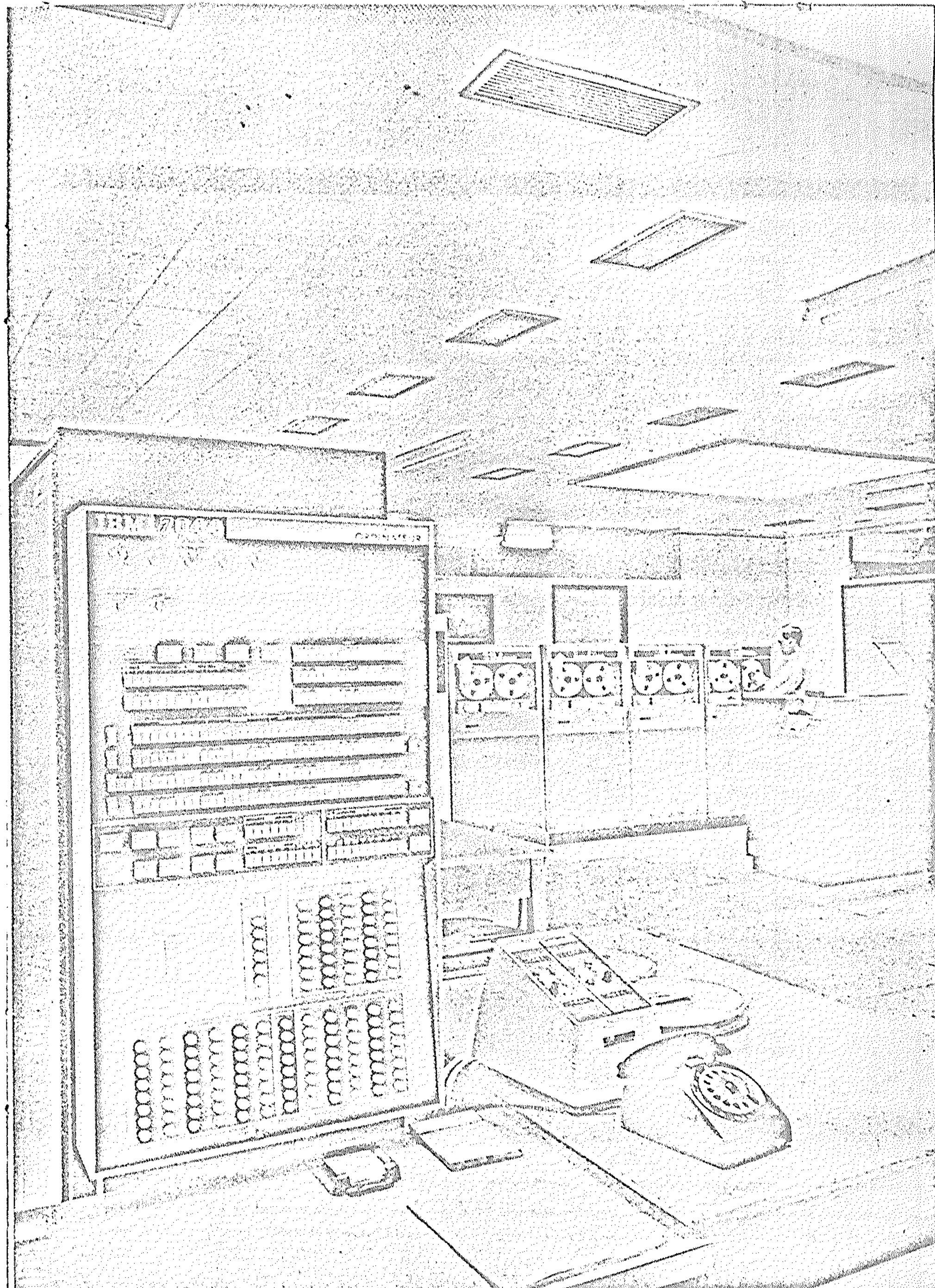
LOGIQUE ET CONCEPTION DES CALCULATRICES

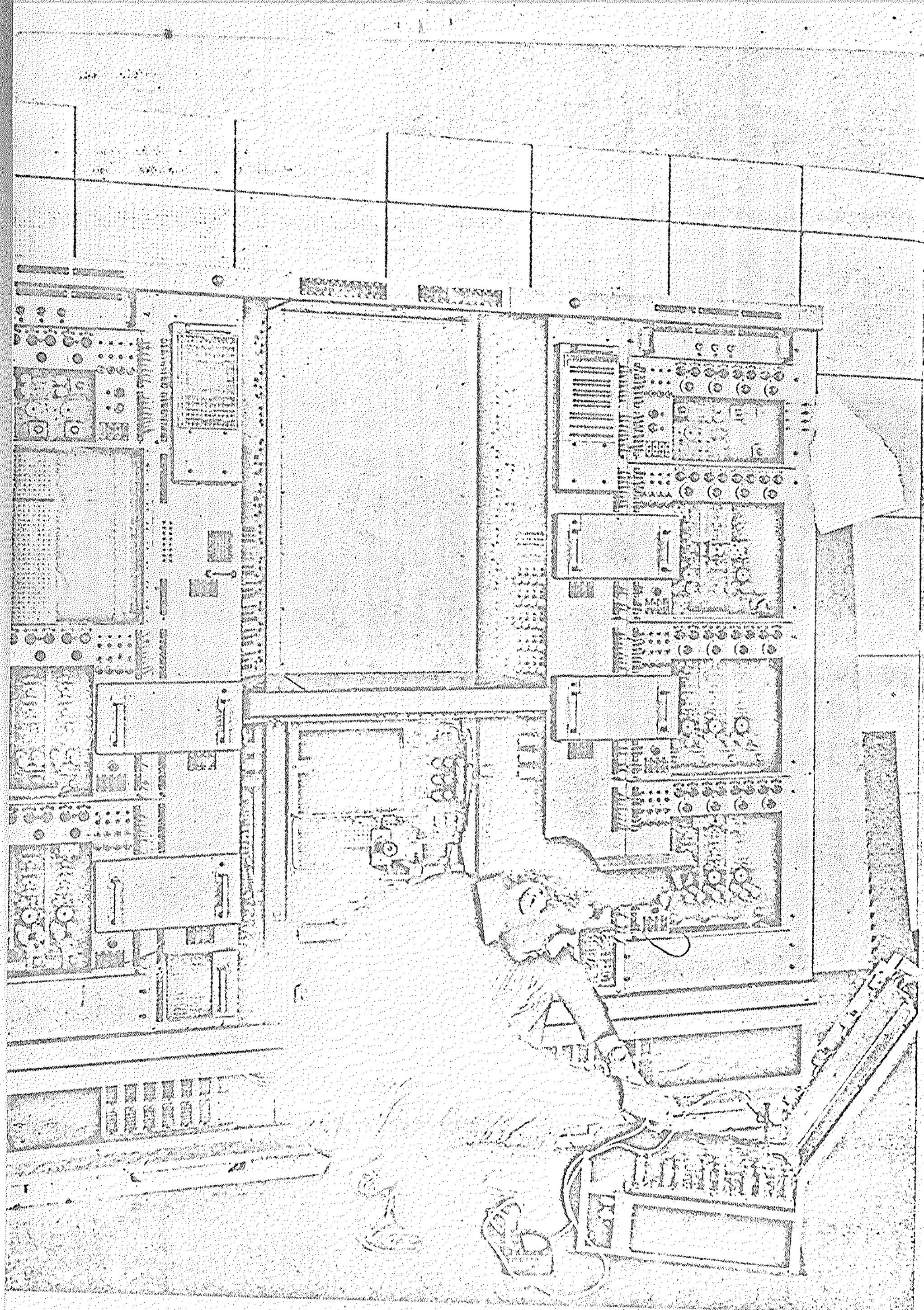
Algèbre logique.
Théorie de l'information.
Analyse de la variance et plans d'expériences.
Physique.
Electronique (circuits intégrés, mémoires)
Logique des systèmes.
Fiabilité.
Technique analogique.
Automatisme et séquence.
Compléments techniques.
Travaux pratiques.

OPTION :

GESTION AUTOMATIQUE

Programmation linéaire.
Programmation dynamique.
Analyse de la variance et plans d'expériences.
Traitement de l'information.
Langage de gestion.
Techniques d'organisation et de production.
Techniques économique, financière et commerciale.
Gestion automatisée.
Simulation.
Jeux d'entreprise.





SECTION SPECIALE

Dans le vaste éventail de matières enseignées à la section normale, la direction conseille à chaque étudiant un ensemble de cours adapté à ses connaissances antérieures et à l'option choisie.

Les étudiants effectuent en cours d'année un voyage d'une semaine consacré à la visite de centres de calcul et d'entreprises de construction de matériel. Ils font en outre un stage d'un mois dans l'industrie.

2. — MAITRISE D'INFORMATIQUE

COMPOSITION

— C 1 : Mathématiques et algorithmique.
— C 2 : Informatique générale.
— C 3 : Logique et programmation des calculateurs ou approximation et optimisation.
— C 4 : Un certificat d'études supérieures au choix du candidat.

HORAIRES, EXAMENS, PROGRAMMES

I. — C 1 : *Mathématiques et algorithmique.*

1° Horaires (*maxima*).

Enseignement théorique : quatre heures.
Enseignement dirigé : deux heures.
Enseignement pratique : quatre heures.

2° Epreuves.

Une épreuve écrite notée de 0 à 20 (durée : quatre heures; coefficient 2).
Une épreuve pratique notée de 0 à 20 (coefficient 1).
L'ensemble des épreuves écrite et pratique est éliminatoire.
Une épreuve orale notée de 0 à 20 (coefficient 1).

3° Programme.

Initiation à l'analyse :

— Calcul intégral.
— Introduction aux espaces vectoriels normés.
— Equations différentielles et fonctions spéciales.
— Distributions, convolution, transformée de Fourier.
— Equations aux dérivées partielles de la physique mathématique.
— Introduction aux méthodes variationnelles.

Probabilités :

— Espaces de probabilités.
— Variables aléatoires réelles.
— Différentes notions de convergence.
— Vecteurs aléatoires dans R^n.

Algorithmique numérique :

- Approximations.
- Quadratures.
- Dérivations.
- Systèmes linéaires d'équations ou d'inéquations.
- Equations algébriques.
- Problèmes différentiels de conditions initiales.

Enseignement pratique.

Algol :

- Pratique des méthodes numériques.
- Pratique du calcul des probabilités.

II. — C 2 : **Informatique générale.**

1° Horaires (maxima).

Enseignement théorique : trois heures.
Enseignement dirigé : deux heures.
Enseignement pratique : cinq heures.

2° Epreuves.

Une épreuve écrite notée de 0 à 20 (durée : quatre heures; coefficient 2).

Une épreuve pratique notée de 0 à 20 (coefficient 1).

L'ensemble des épreuves écrite et pratique est éliminatoire.

Une épreuve orale notée de 0 à 20 (coefficient 1).

3° Programme.

Notions de théorie de l'information, codage.

Moyens et méthodes de traitement de l'information sous sa forme quantifiée :

- Théorie du signal. Applications au fonctionnement des opérateurs logiques et arithmétiques et au stockage de l'information.
- Notion d'automates séquentiels.
- Théorie des algorithmes. Application à la programmation, aux langages et aux systèmes.

Traitement de l'information sous sa forme continue :

- Notion d'opérateurs analogiques.
- Principes de fonctionnement et méthodes d'utilisation des machines informatiques analogiques et d'ensembles de traitement répétitif ou hybride.
- Méthodes d'optimisation.

Introduction à l'étude physique des machines informatiques :

- Notions de physique du solide; applications des composants électroniques semi-conducteurs, supraconducteurs, ou magnétiques aux circuits des machines informatiques.
- Electronique impulsionnelle.
- Méthodes d'amplification à grand grain. Systèmes bouclés.
- Notion de fiabilité.

III. — C 3 : A) **Logique et programmation des calculateurs.**

1° Horaires (maxima).

Enseignement théorique : trois heures.
Enseignement dirigé : deux heures.
Enseignement pratique : cinq heures.

2° Epreuves.

Une épreuve écrite notée de 0 à 20 (durée : quatre heures; coefficient 2).

Une épreuve pratique notée de 0 à 20 (coefficient 1).

L'ensemble des épreuves écrite et pratique est éliminatoire.

Une épreuve orale notée de 0 à 20 (coefficient 1).

3° Programme.

Compléments d'algèbre :

- Algèbre de Boole.
- Monoïdes.
- Automates finis.
- Eléments de logique formelle et de calculabilité.

Théorie des langages :

- Définition des structures d'information, listes, manipulation de ces structures.
- Analyse détaillée des principaux langages (Algol, Cobol, LISP).
- Analyse syntaxique des langages. Compilation.

Théorie des circuits :

- Application de l'algèbre de Boole à la synthèse des circuits.
- Circuits séquentiels (aléas).
- Détection d'erreurs et codes correcteurs.

C 3 : B) **Approximation et optimisation.**

1° Horaires (maxima).

Enseignement théorique : trois heures.
Enseignement dirigé : deux heures.
Enseignement pratique : cinq heures.

2° Epreuves.

Une épreuve écrite notée de 0 à 20 (durée : quatre heures; coefficient 2).

Une épreuve pratique notée de 0 à 20 (coefficient 1).

L'ensemble des épreuves écrite et pratique est éliminatoire.

Une épreuve orale notée de 0 à 20 (coefficient 1).

3° Programme.

— Compléments de mathématiques (analyse fonctionnelle).
— Itération (récurrence, gradient, théorème du point fixe et applications.
— Discrétisation (équations fonctionnelles en général).
— Approximation (caractérisation et construction d'approximations, convergence, stabilité).
— Programmation mathématique (linéaire, non linéaire, dynamique, théorie de Pontryagin).

Enseignement pratique.

— Mise en équation de problèmes concrets.
— Techniques de calcul.
— Mise en œuvre des méthodes sur calculateurs.

3. — MAITRISE DE MATHÉMATIQUES
ET APPLICATIONS FONDAMENTALES

COMPOSITION

— C 1 : Calcul différentiel.
— C 2 : Calcul intégral.
— C 3 : Mécanique analytique et mécanique des milieux continus, ou analyse numérique, ou algèbre et [illegible], ou pratique fondamentale.
— C 4 : Un certificat d'études supérieures au choix du candidat.

HORAIRES, EXAMENS, PROGRAMMES

I. — C 1 : Calcul différentiel.
— C 2 : Calcul intégral.

[illegible]

Une épreuve orale notée de 0 à 20 (coefficient 2).

3° Programme.

Mécanique analytique :

— Formalismes lagrangien et hamiltonien. Principes variationnels.
— Théorème de [illegible].
— Théorème de Poisson-Liouville.
— Équation de Jacobi.

L'enseignement de mécanique analytique doit préparer les développements ultérieurs de mécanique quantique, de mécanique statistique ou de mécanique céleste. Le professeur pourra, s'il le désire, introduire des méthodes de géométrie différentielle moderne.

Mécanique des milieux continus :

— Cinématique. Déformations. Contraintes.
— Équations générales des milieux continus.
— Formulations intégrale et différentielle des lois de conservation. Discontinuités.
— Éléments de thermodynamique des milieux continus. Bilan énergétique.
— Lois de comportement : milieux élastiques, fluides parfaits et visqueux, autres exemples.
— Linéarisation; étude complète de quelques problèmes linéaires d'élasticité et de mécanique des fluides. Acoustique.

C 3. 2) Analyse numérique :

1° Horaires (maxima).

[illegible] trois heures.

[illegible] heures.

[illegible] deux heures.

[illegible]