A DIVERSIFIED UNDERGRADUATE COMPUTER SCIENCE PROGRAM

Richard Cheng
Department of Computer Science & Technology
Rochester Institute of Technology
Rochester, New York

1. Introduction

This paper introduces a set of unique undergraduate computer science programs, some of which have been operating for more than two years at Rochester Institute of Technology. These programs are intended to meet current computer science manpower demands where as a generalized computer science curriculum is not sufficient to cover the range of students required by such demands. As is shown in Figure 1, the set of programs extend from a general computer science curriculum to cover several quasi-specialized areas. Each area is offered as a curriculum that will be defined and elaborated on in the sections that will follow.

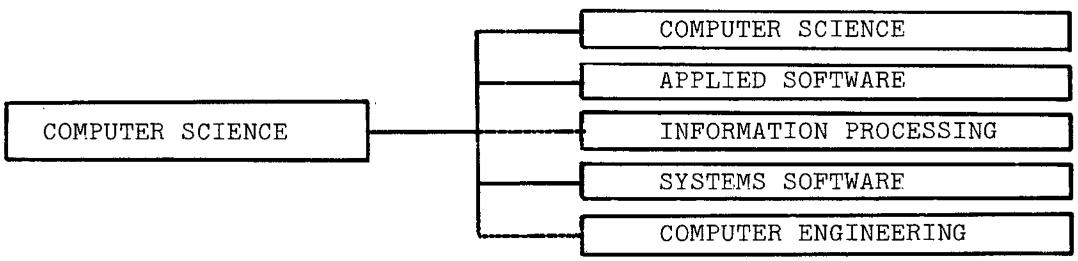

Figure 1. Computer Science Program Structure at RIT

The structure in Figure 1 does not reflect the administrative organization at RIT. However, these curricula are offered by the Department of Computer Science and Technology with the exception of Computer Engineering, which is jointly offered by the Electrical Engineering Department.

2. Objectives and Rationale

Generally, undergraduate curricula are designed to (a) prepare students for graduate studies and (b) prepare students for employment. While all of the five RIT undergraduate Computer Science curricula are providing sufficient backgrounds for students to pursue graduate studies, the main objective of the four quasi-specialized curricula is to train students in areas where high employment demands exist. The general Computer Science curriculum leads to more theoretical studies, such as theory of computation and automata theory, and provides students with flexibility for future studies and specialization may be obtained through the selection of a special area in graduate studies. Table 1 indicates training fields and job descriptions of graduates from each of the four quasi-specialized curriculum.

TABLE 1. Field and Job Title for Graduates

Curriculum	Field	Job
APPLIED SOFTWARE SCIENCE	SCIENTIFIC	RESEARCH PROGRAMMER, APPLICATIONS SPECIALIST, MATHEMATICAL COMPUTATION PROGRAMMER, LANGUAGE PROCESSOR DESIGNER
SYSTEM SOFTWARE SCIENCE	SYSTEMS	SYSTEMS PROGRAMMER, DIAGNOSTICS SPECIALIST, SYSTEM SOFTWARE DESIGNER, TELECOMMUNICATIONS SOFTWARE DESIGNER
INFORMATION PROCESSING	BUSINESS	APPLICATIONS PROGRAMMER, SYSTEMS ANALYST, DATA PROCESSING SPECIALIST
COMPUTER ENGINEERING	HARDWARE	COMPUTER DESIGNER, PROCESS CONTROL ENGINEER

A. APPLIED SOFTWARE SCIENCE

There are demands for scientific application programmers in industry, government, research and educational institutions. Currently, the scientific programmers' positions are mostly filled by people trained in a scientific discipline who have taken one or two programming courses or have learned programming through practice on jobs. General computer science undergraduates are capable of taking on such jobs after some additional or on the job tarining. Graduates from the Applied Software Science curriculum, however, will have the advantage of selecting a specific scientific discipline as science electives and taking computer science courses designed for scientific applications. Upon graduation, the applied software scientist will be productive without requiring further training.

In the Applied Software Science curriculum core courses, FORTRAN programming, Numerical Methods, Simulation, Real-Time Computation, Minicomputer Systems and Applications provide the students with an essential background in computer application techniques for science areas they have chosen. Projects are encouraged in this curriculum as electives to enhance experience in selected areas. Equipment provided for this curriculum includes an SEL810B computer, SEL Data Aquisition System, IBM 1500 computer system and a PDP 11/10 computer.

B. SYSTEM SOFTWARE SCIENCE

System programmers are needed in medium to large scale installations. In a general Computer Science curriculum, there is very little emphasis on the applied aspect of systems software. Therefore, system programmers today have been promoted from experienced applications programmers and the supply is thus far below the demand. Since systems programming requires extensive and specific techniques and experience instead of theories, a special curriculum must be designed to train systems programmers.

In the System Software Science curriculum Assembly Language and Advanced Assembly Language Programming are essential fundamentals. The core courses include Operating Systems; Assemblers, Interpreters and Compilers; System Programming; Systems Programming Lab; Microprogramming and Minicomputer Systems. The above courses provide students in this curriculum with sufficient training in systems programming techniques in general. A follow-up of one or two quarters of systems programming projects provides students with more specific experience in systems software. In this curriculum hands-on experience on a large scale computer and minicomputer system is provided, and is manditory. Currently, an IBM System 360/30F, a PDP 11/40, a Microdata 1600D and an Interdata 7/16 are used for such purposes.

C. INFORMATION PROCESSING

The most demanded computer oriented personnel currently is in data processing industry. However, this area has been largely ignored by Computer Science education programs throughout the nation. The data processing training has been left to two-year community colleges and Business curricula. As the data processing techniques have become more sophisticated, the trend has shifted to employing four year graduates in the Computer Science curricula. The Information Processign curriculum is designed to produce Business Applications Programmers and Systems Analysts. The core courses include COBOL, Advanced COBOL Programming, Information Systems Design, Information Systems Analysis, Data Communication Systems, Computing Management, Data Base Concepts, and electives. These courses and systems work shops provide students with essential backgrounds in business applications and systems analysis. In addition, students in this curriculum normally take 44 quarter credits of minor studies in the Business curricula. In this curriculum, application experience is provided on medium and larger scale computers with magnetic tape and disk file facilities. This includes an IBM 360/30F and an XDS Sigma 6.

D. COMPUTER ENGINEERING

Computer Engineering is a hardware oriented curriculum jointly offered by the Computer Science and Electrical Engineering Departments where demand comes from computer manufacturers and process control industries. Core courses include Electronics, Logical Design, Microprogramming and Digital Data Communications. Equipment provided in this curriculum includes an HP-2100 system and an Intel Model 8 system.

3. Curriculum Description and Sample Programs

All RIT undergraduate degree programs are five year programs, one year of which is devoted to required cooperative work experience. Starting in the junior year, a student alternately works for industry full-time for one quarter and studies full-time for another quarter. The total required coop work experience is four quarters as partial fulfillment for a baccalaureate degree. The Bachelor of Science (B.S.) degree requires 50% of Liberal Arts credit and 50% professional credits, while the Bachelor of Technology (B.T.) degree requires 25% Liberal Arts credit and 75% professional credits. At RIT, the quarter system is practiced, i.e. a quarter credit equals two thirds of a semester credit. In the following sample curricula, quarter credits are used.

All Bachelor of Technology curricula require a minor, such as Business (Information Processing majors), Mathematics, Engineering or Physical Sciences. The Bachelor of Science curricula do not require a minor. A simplified breakdown of degree requirements for B.S. and B.T. degrees is shown in Table 2. Course structure and key courses for each curriculum are shown in Figure 2. Numbers in boxes are course numbers. Corresponding course titles can be found in the Appendix. Sample programs for each curriculum are shown in the RIT catalog.

TABLE 2. Degree Requirements

B.S. DEGREE IN COMPUTER SCIENCE CURRICULUM

Area of Study	Quarter Credits
Computer Science	88
Mathematics	24
Math/Science	24
General Studies	54
Free Electives	8
TOTAL	198

B.T. DEGREE IN COMPUTER TECHNOLOGY CURRICULUM

Area of Study	Quarter Credits
Computer Science	92
Mathematics	11
Minor	44
General Studies	47
TOTAL	194

4. Conclusion

The diversified undergraduate program provides students with opportunities to pursue generalized Computer Science for future studies or to pursue a specialized area for employment following graduation. Responses from students at RIT are most favorable. Currently, at RIT the Information Processing program has the largest enrollment which reflects the local industrial demands and the fact that it has been in operation for five years. System Software Science and Applied Software Science have a combined enrollment of approximately one half the number of Computer Systems students. Computer Science and Computer Engineering have the smallest enrollments. These two programs are the newest ones and their enrollments indicate that students at RIT are very sensitive to the trend of manpower demands. Most of the Information Processing graduates (the only program that has graduated students) have sought employment. Some graduates have applied for admission to graduate schools and have been accepted by schools such as Rensselaer Polytechnic Institute and MIT.

Responses from industrial employers have been most encouraging. The following facts indicate our experience with industries that employ our coop or graduated students. (a) All June, 1975 graduates who seeked employment have been placed. (b) Coop employers have come back to RIT to ask for more coop students in specified Computer Science areas. (c) The popularity, based on frequency of inquiries by industrial employers, is in the following order: Information Processing, System Software Science and Applied Software Science. Computer Science and Computer Engineering are too new to provide useful data. (d) The local industrial advisory group representing eleven different industries have endorsed the diversified approach to Computer Science education based on their employment preference.

The only problem associated with this program is the name of these Computer Science areas. High school and transfer applicants are confused by the names in the RIT catalog. Our staff in the Admission's Office and in the Department have repeatedly answered questions of that nature to explain what each area is all about. A very positive result of this diversified approach is the reduction of the overall attrition rate from 17% before the change to 6% at present. There have been many cases of migration within the Computer Science areas in the Department instead of dropping out entirely from the Computer Science curriculum.

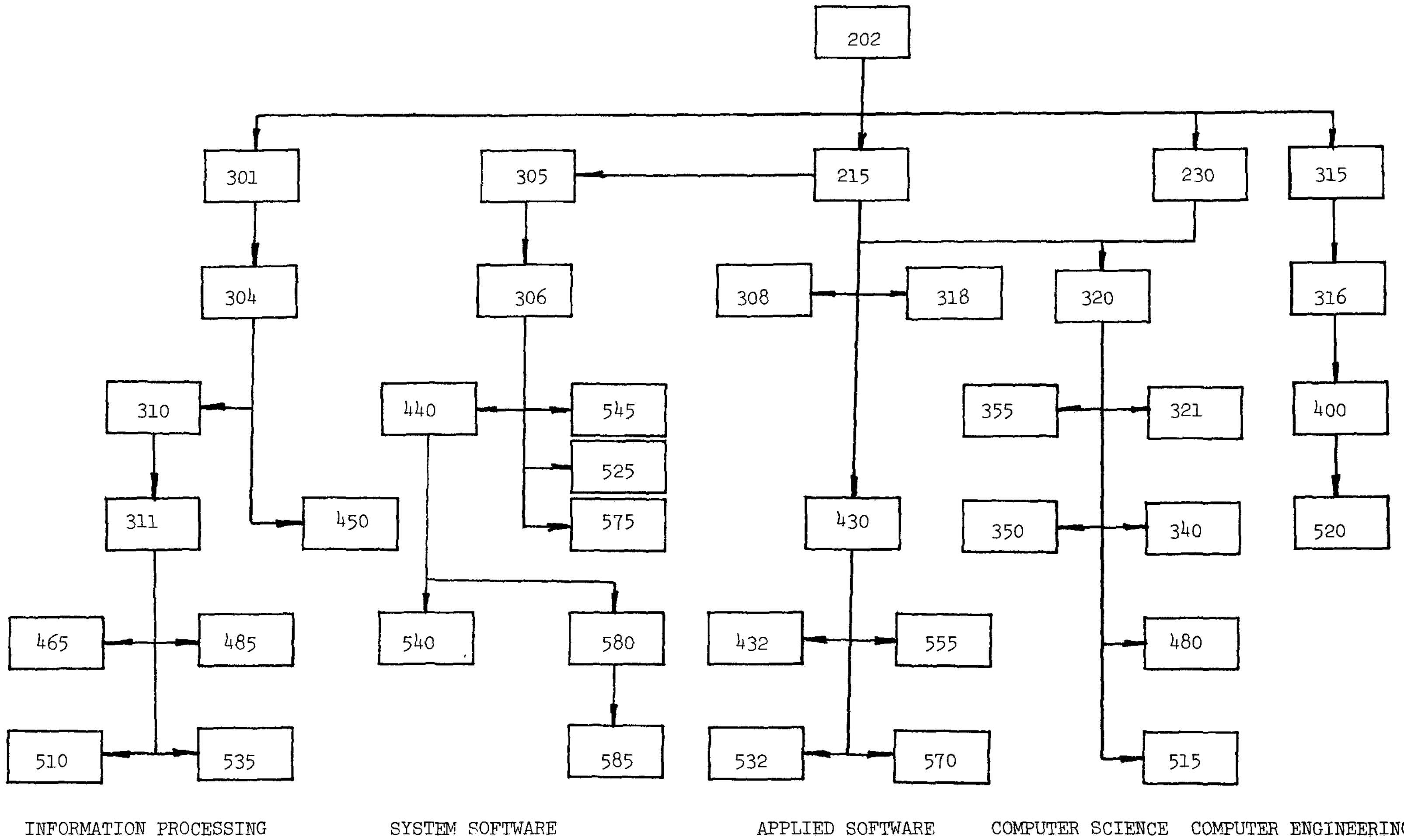

Figure 2. COURSE STRUCTURE AND KEY COURSES IN COMPUTER SCIENCE AND TECHNOLOGY PROGRAMS

APPENDIX

COURSE LIST

COURSE NUMBER	COURSE TITLE
ICSS202	Introduction to Computer Science
ICSP215	Programming Language - FORTRAN
ICSS230	Discrete Structure
ICSP301	COBOL Programming
ICSP304	Advanced COBOL Programming
ICSP305	Assembly Language Programming
ICSP306	Advanced Assembly Language
ICSP308	Structured Programming
ICSP310	Information Systems Design
ICSS311	Information Systems Analysis
ICSS315	Digital Computer Organization
ICSS316	Introduction to I/O Systems
ICSP318	APL Programming Techniques and Applications
ICSS320	Data Structure Analysis
ICSS321	Sorting and Searching Techniques
ICSS340	Finite State Machines and Automata
ICSP350	Programming Language Concepts
ICSS355	The Human Side of Computers
ICSS400	Logical Design
ICSS420	Data Communication Systems
ICSS430	Numerical Methods
ICSP432	Computer Application in Analysis and Design
ICSS440	Operating Systems
ICSS450	Computing Management
ICSS465	Introduction to Management Information Systems
ICSS480	Formal Language
ICSS485	Data Base Concepts
ICSS510	Systems Workshop
ICSS515	Analysis of Algorithms
ICSS525	Assemblers, Interpreters and Compilers
ICSS530	Discrete Simulation
ICSP532	Computer Application in Social & Behavioral Science
ICSS535	On-line Information Systems Design
ICSS540	Operating Systems Laboratory
ICSS545	Microprogramming
ICSS555	Real-time Computation
ICSS570	Computer Graphics
ICSS575	Minicomputer Systems and Applications
ICSS580	Systems Programming
ICSS585	Systems Programming Laboratory
ICSS520	Computer Architecture