Computing at the junior/community college—
Programs and problems

by HAROLD JOSEPH HIGHLAND
State University Agricultural and Technical College
Farmingdale, New York

INTRODUCTION

This and the following papers contain different views of the same subject—“Computing Education at the Junior/Community College.” It is a topic that has been long neglected, swept under the rug so to speak, in computing circles. It is about time that this topic was fully aired and its vital importance recognized by all engaged in computer education, business, industry and government. There are probably more students and more teachers involved in computer education at the junior/community colleges than at any other level of education.

Before proceeding, I should like to thank the participants of this panel for their enthusiastic cooperation and valuable contribution. Although they represent different parts of the country and different programs, they are all involved with junior/community college computer education.

• Dr. Alton Ashworth of Central Texas College (Killeen, Texas) has been in charge of developing a model program for the junior/community college under an Office of Education grant.
• Mr. Robert G. Bise of Orange Coast College (Costa Mesa, California) developed the prime program in computing education at the junior/community college-level, which has served as the prototype for such programs in California.
• Professor Donald Davidson of LaGuardia Community College of the City University of New York (Long Island City, New York) has been instrumental in developing a work-study program for underprivileged students in a metropolis.
• Dr. John Maniotes of the Calumet Campus of Purdue University (Hammond, Indiana) has had extensive experience in developing and running an integrated two-year and four-year program in computing on his campus.
• Professor Charles B. Thompson of New York State University Agricultural and Technical College at Farmingdale (Farmingdale, New York) has been very instrumental in the development of a dual program designed not only to meet the needs of career-oriented students but also one to serve students who plan to continue their education in this field at a four-year college.

Furthermore, I should like to define, or perhaps explain, the use of the term, "academic computing in the junior/community college." It was selected, not because we need to add to the myriad of terms we already have in computer education, but because there was no term broad enough to cover all aspects of computing education at this level of higher education.

• In some institutions, the term, computer science, is used but many times the courses and the level at which they are taught bear no relationship to computer science taught at a four-year college, following the guidelines of Curriculum ’68 which was developed under Dr. William F. Atchison.
• In other institutions, the term, data processing, is used; but here again there are extremely wide variations. Not all such programs are solely and purely business-oriented.
• The term, computer technology, is likewise encountered at the junior/community college. Some of these programs are designed to educate electronic technicians; others involve the training of computer operators. Still others more closely resemble computer science at the four-year college or data processing in a college of business administration.
• Finally, we are beginning to encounter the term, information processing, since curriculum titles are being used at times to show that one is keeping up with the state of the art. Oftentimes, the courses and their content are far different from the program proposed by the ACM Curriculum Committee on Computer Education for Management (C’CM) for undergraduate education under the leadership of Dr. J. Daniel Couger.

JUNIOR/COMMUNITY COLLEGE PROGRAMS

Having served as a director of a graduate business school as well as a dean of instruction at a four-year liberal arts college, I was startled when I joined a two-year
From the collection of the Computer History Museum (www.computerhistory.org)

college to develop a program in computing. The lack of uniformity in course selection, course sequence, proportion of theory and application taught, were immediately evident. Programs were being developed all over the country and often were designed to meet immediate near-term needs or merely to be "modern." Little or no concern was given to the impact of technology and the possibility of obsolescence of the graduates from these programs. One of my associates engaged at the graduate level in computing assured me that diversity meant freedom.

Yet, I see this diversity as chaos with too many charlatans and humbugs performing rituals in hexidecimal in the classroom without concern for the quality of education or the future of their students, or sincere educators who cannot upgrade the quality of their educational programs because they cannot provide their administration with "authoritative, professional guidelines." Now, many years later, I find that some of the extremes have died but there is still a lack of cohesion in computing programs at the junior/community college level. Let me just touch several of these areas.

**Department structure and affiliation**

Where is computing taught at the junior/community college level? In some schools there is a separate department of computer science, data processing, computer technology; a department which sets its own curriculum and guidelines. In other institutions, computing is part of the business department; it is one more 'major' in the class of marketing, accounting or management. Still at other colleges, computing is part of the mathematics department; here we most often find that curriculum which is closest to the four-year college in computer science. Yet, the emphasis is primarily a mathematical approach without concern for computing applications in other areas.

**Faculty education and experience**

Because of the rapid growth in the number of junior/community colleges over the past several years and the increased demand for computer faculty at four-year and graduate schools, the junior/community colleges have been low man on the totem pole. Except for a core of dedicated teachers, most of those with adequate education and experience have not, until recently, been attracted to the junior/community colleges. At a level where application is emphasized at the expense of theory, we find many teachers who have never had practical experience in computing in industry, business and/or government. Too many teach from the textbook or repeat what they have learned from their professors at four-year and graduate schools, and many of them as well have spent all their years in the ivory tower.

**Programs and options**

Depending upon the individual school, the student interested in computing may be forced into some narrow area, such as business data processing. Many of the junior/community colleges are too small to offer a broad spectrum of computing courses. The areas in which they train students include:

- keypunch operators
- computer operators
- computer programmers.

In some schools, the computing programs are career-oriented, and except in few cases, these students find that their two years of education is discounted if they wish to continue their education at a four-year college. In other schools, computing is computer science oriented and the student wishing to work upon graduation does not possess the necessary skills to find and hold a job.

The problem of training computer operators is a critical one at the junior/community college level. Too many of the schools have inadequate computing facilities to provide a proper program in this area. Some have turned to simulators to do the job. Any of you who have had experience with most of these simulators recognize their numerous shortcomings. (I should apologize to my colleagues in the simulation area for that comment since I am editor of the ACM SIGSIM Simuletter, but in this case, I feel that it is the truth.) Other schools have turned to work-study programs or cooperative training programs wherein the students study the theory of operations in school but obtain their experience at cooperating companies in industry and business.

**Computer courses and sequence**

In this stage of computing, can anyone imagine spending time in a one-year course in "electric accounting machines?" Yet, there are a number of two-year schools, both public and private, that train their students in unit record equipment and spend considerable time in wiring. At the other end of the spectrum are the schools which require career-oriented students to take courses in logic circuits, Boolean algebra, and hardware specifications. In fact, until recently, there was one school which spent one half of a one semester course teaching keypunching to students who supposedly were being trained to become junior programmers.

Where does a course in systems analysis fit into the curriculum? Is this one which is taught in the first semester concurrent with an introduction to data processing, or is this the capstone in which students can utilize the information they learned in all their other courses? Similarly, should the students be required to take statistics with the mathematics department and do their work with pencil and paper or even a calculator, or should they use
the computer and spend less time on the mechanics and more on the concepts?

Credits in computing

How many credits in a two-year program should be devoted to computing? Currently, there are schools that offer a data processing “major” with as little as 12 credits in computing (and six of these are in electric accounting machines) to schools which require almost 49 credits out of a total of 62 to 64. What is the minimum number of courses and/or credits which should be taught? And which courses?

Computing facilities

Many of the junior/community colleges have some computing facilities available for student use. Yet there are some offering computing programs that do not have any computing facility available for student use. One cannot but question the value of such a program.

Furthermore, what type of facilities are available for which programs? Do you need the same for computer science (in the four-year sense) as you do for a career-oriented program in the business area?

It is possible to continue in examining other areas of diversity, but it should be apparent that there is a wide spectrum of programs under the heading of computing in the junior/community college.

SOME PROBLEMS TO BE ANSWERED

The two-year college, no matter what it is called (junior or community or as has become fashionable to leave either term out), is a unique institution in education. In some cases its students vary little from those who enter a four-year institution, but in other cases, these two-year schools receive those students who cannot be admitted to the four-year colleges.

Computing languages

The number and intensity of languages studied varies greatly among the junior/community colleges. There is also great variation in which language is taught first and in what sequence are languages studied by the student. Among the languages offered by the two-year colleges are: BASIC, COBOL, FORTRAN, RPG, PL/I, APL, AL, JCL. At some schools students are required to take only one language during their entire two years, while in a few three and even four languages are taught to all students as part of the basic program.

At this level of instruction, which is the simplest language to introduce to the students? Some look upon BASIC as being too much like FORTRAN and therefore too scientific, unsuitable to students going into the business field. Many schools start with FORTRAN, but in one, a year of COBOL is the prerequisite for the study of FORTRAN. A few, like my own college, start with PL/I.

Since these schools are more often under local community control as compared with four-year colleges and universities, the programs should be designed to meet community needs. But a broader view is also necessary. It is about time that we recognized that the four-year colleges in computing are almost monolithic in their programs as compared with the two-year schools. The computing community has an obligation to see that some level of competency or adequacy is set for these institutions. I am not proposing a system of accreditation but the establishment of some guidelines, fundamental curriculums to meet several purposes.

Attention should also be devoted to the high level of attrition in these programs. Is it really the student’s fault that they have failed? Or is it the lack of a properly sequenced curriculum, adequate teaching aids, quality of the teachers, or what?

Many teachers and administrators at the junior/community college level require some professional ammunition in attempting to get college presidents, local appropriation committee, etc., to upgrade existing equipment and programs. It is here that ACM can play a strong role, but it must be done now.

In addition, a greater exchange of information among the junior colleges is necessary. An exchange of curriculums, course outlines, booklists—an airing of problems: how much mathematics should a student be required to take, what techniques have been used to cut attrition, what arrangements have been made for transfer of students—is essential in this area.

It appears apparent that in light of accelerated computer technology, the limited computing facilities at the junior/community college level and concomitant problems that many of the two-year programs offer today will not be viable within the near future. Computing is already making inroads at the secondary school level. In some parts of the country, and this we have locally, there are special vocational educational centers for intensive training of high school students. If they develop adequate programs for training input clerks, control clerks and computer operators (and eventually they will), what will be taught at the two-year level?

Finally, to do its job effectively, the junior/community college must have better information about industry’s and government’s needs in the computing field. Today, the Bureau of Labor Statistics either lumps all those engaged in computing into a single category or at most separates programmers from the rest. What can be done to obtain greater insight into these needs so that more effective programs can be developed and taught at the junior/community college level?

The problems are many and those who are truly interested in doing are few. Those of us within ACM should seek some dynamic structure within which to operate; now is the time to start.