Introductory Service Courses in the Computer Science Curriculum

by J. M. Adams and D. H. Haden
Computer Science Department
New Mexico State University

I. INTRODUCTION

The organization of the computer science curriculum to adequately meet the service needs of other departments while still providing a reasonable program for majors is at best difficult, but is especially difficult within the economic constraints of relatively small universities. This paper contains some thoughts on this difficulty, beginning with an attempt to delineate the service needs and the associated problem areas and then making some rather general curricular recommendations for introductory service courses. Finally, experience with one of the most important recommendations is discussed in detail in the hope that it will form some sort of crude model for other developments.

II. NEEDS AND PROBLEMS

After some obvious consolidations, computer science educational service needs can be categorized along disciplinary lines as follows:

(1) Natural Sciences and Engineering
(2) Behavioral Sciences
(3) Management Sciences
(4) Liberal Arts
(5) Teacher Education

Naturally, it would be quite simple, and probably effective, if course sequences could be offered for each of these areas. Since this is not economically feasible for many institutions, one searches for a more consolidated classification. A simple breakdown of students into those who wish to learn to use the computer and those who wish some general appreciation of computers and the implications of their usage leads to a very compact classification which is the basis for many service offerings.

The appropriate classification for any one institution is probably some mixture of the two mentioned above, and any attempt to designate a single "best" classification scheme for all institutions would be unreasonable. However, the two classifications given above provide a basis for identifying problem areas.

Among students who wish to learn to use the computer, their disciplinary association gives rise to the following problems:

1. Level disparity
2. Background disparity (primarily mathematical)
3. Application disparity

To illustrate these problems, let us consider two disciplines characterized as follows:

Discipline 1
1. Courses involving use of the computer start at the sophomore level.
2. Majors must meet mathematics entrance requirements for science and engineering.
3. Computer applications in the discipline are primarily of a numeric scientific nature.

Discipline 2
1. Courses involving use of the computer at the senior-graduate level only.
2. Minimum mathematics background required of majors.
3. Computers applied mainly as a device to handle large data files.

The characterizations of the disciplines create the disparities mentioned before. Attempting to serve the needs of even these two disciplines with a single introductory course is naturally a nightmare due to the heterogeneous nature of the classes.

For students interested in obtaining a general appreciation of computers and the implications of their usage, the disparity problems are not quite so serious. It has been our experience that even though a mathematical background disparity may exist, it can usually be accommodated as can an application disparity. Level disparity has proven to be the most serious of the three since the more mature student has a need for deeper investigation and freer discussion.

Having identified some of the problem areas, we turn in the next section to recommendations for alleviating these problems without excessive course proliferation.

III. RECOMMENDATIONS

In an introductory "usage", as opposed to an "appreciation-implications" course, a solution to both the background and application disparity problems can be approximated by either splitting the course into two courses or merely giving different orientations to different sections of the same course. Even though the same programming language may be taught, the examples and exercises for one orientation may be quite quantitative and scientific, while those for the other orientation may be far less quantitative but far more concerned with the manipulation of large data files.

The level disparity in introductory usage courses is somewhat more difficult to deal with, but one solution may be based on the substitution of intensity of material for depth of material. For example, two courses covering virtually the same material may be offered at different levels with the higher level course carrying less credit than the lower level course. At New Mexico State the higher level course is one credit hour at the senior level and the lower level course is three credit hours at the freshmen level.* Naturally there are fundamental differences in the manner of conducting the courses since students in the higher level course meet less frequently and are directed in a self-study approach by the instructor.

Since the background and application disparity problems can be accommodated in introductory appreciation-implications courses, a single course may suffice. Furthermore, if a single lower level appreciation-implications course can be effectively taught to large classes, the economic advantages are quite significant. The level disparity still poses enough of a problem to warrant the same type of solution proposed for the introductory usage courses. That is to say a higher level, lower credit course can be offered for more advanced students.

Since the combined lower level introductory appreciation-implications course offers by far the most significant economic advantages, some of our experiences with such a course will be detailed in the next section. Many of the observations made previously have arisen from our experience with this course.

IV. EXPERIENCE WITH ONE COURSE

Introduction to Computers, a three credit hour freshman course at New Mexico State, has the following course description:

> A basic general course with a non-technical emphasis. The evolution of computers, the applications of computers, and their economic and social implications. A brief introduction to programming.

A current syllabus for the course is given below:

Number of Lectures	Subject
1	Introduction to the course
4	Introduction to the computer and short history of computers
4	Introduction to BASIC Programming
1	First Exam
2	Computer Applications I (scientific, industrial, business, decision-making and large-data-base applications; more BASIC programming)
1	Second Exam
2	History of Computation
2	Machine-Level Programming
3	Computer Hardware
3	Computer Languages (survey)
1	Third Exam
8	Social and Economic Implications (invasion of privacy, unemployment, leisure, life in a mechanized world, etc.)
2	Computer Applications II (programming non-numeric problems, exotic applications: artificial intelligence, language translation, future prospects, etc.)
1	Fourth Exam
1	Continuation of Computer Applications II
1	Review
1	Comprehensive Final Examination

*The lower level course is also the first course taken by computer science majors.

It would have been nice to have had the course spring forth full blown in its present form but of course this was not the case. The present course is the result of the evolution of a course first taught five years ago. Many failures preceded what the authors now consider a relatively successful course.

The course was already on its way to becoming a large lecture session five years ago. The student profile was quite heterogeneous in age, major, classification, background knowledge, and general ability. The disparities outlined previously began to be identified at that time.

As the course enrollment increased (to 300-500 per semester) effort was expended in four primary areas: (1) obtaining adequate materials for the course, (2) arranging the mechanics of the course to accommodate the increasing number of students, (3) providing the students with an adequate programming environment, and (4) making quantitative assessments of the course.

COURSE MATERIALS

One of the complications produced by the disparities itemized previously is finding materials appropriate for the course. A brief sketch is given here of the authors' experience in resolving complications in this area.

A proper appreciation for the problems associated with an appreciation-implications course began to develop when an attempt was made to obtain text materials with the appropriate content for the course. The need to teach more than programming was apparent and the desirability of discussing the social implications of computing seemed equally apparent. There was also a desire to stress a broad range of applications and their economic implications. No single source of material could be located at that time which met the course-content needs. In an effort to obtain the materials, films, new publications, trade-journal articles, works of fiction, and supplementary notes were employed.

The materials currently used in the course fall into four basic categories:

1. audio/visual
   a. films
   b. tape recordings
   c. slides
2. library readings
3. supplementary handouts
4. main text

Some of the films currently shown in the course are somewhat old but are considered to be excellent for our purposes:

1. "What is a Computer?", Encyclopedia Britanica, Educational Corporation
2. "America: On the Edge of Abundance", Audio-Visual Center, Indiana University
3. "The Living Machine", Contemporary Films, McGraw-Hill Book Company
4. "The Wierd World of Robots", Contemporary Films, McGraw-Hill Book Company
5. "Experiments in Motion, Graphics and Permutations", IBM

Magnetic tape recordings were obtained from the Center for the Study of Democratic Institutions and are a six-part series on the Machine Image.

Library readings were only partially successful. Items tend to wear out or get damaged or stolen before each of 450 students has had an opportunity to read them. In lieu of this, copies of supplementary articles, usually from the local newspaper or news magazines are handed out each semester.

The main text now in use is the end product of about four years effort. It was written by the present authors and sold to students as locally produced notes until last year when Kendall/Hunt produced a limited number of copies of a preliminary version of the text. The text is now under contract to John Wiley and Sons.

The text is fairly evenly divided among programming, applications, history, hardware, and the social implications of computing. In an effort to make the latter topic more interesting to students (and some effort is required to do this), rather diverse materials have been collected (e.g., the Bob Elliot and Ray Goulding play from the Atlantic Monthly, "The Day the Computers Got Waldon Ashenfelter" and about thirty cartoons selected for their relevance in social implications).

One text has been published recently that the authors feel meets many of the course needs: Richard Dorf's, Introduction to Computer Science. Brevity in social implications is the primary disadvantage for the course.

COURSE MECHANICS

A consequence of the consolidation of lower level introductory appreciation-implications courses is, of course, large lecture sections. Course mechanisms have been worked out at New Mexico State over the past several years to facilitate handling these large sections.

The goal of these mechanisms has been to relegate many of the routine matters (attendance, seating, grading) to standard policies and free the instructor's and grader's time for personal attention to exceptional problems. Students, if they understand that this is the goal, seem not to be bothered as much by the depersonalization of the large class as they are when such mechanisms are not present.

Some of the procedures used are standard but have been found to be quite necessary. They include use of seating charts, frequent short quizzes, machine-scored objective exams, computer grade-recording programs and several levels of student programming assistance. A program to generate exams would be very helpful but is not currently available at New Mexico State.

A casual approach to course mechanics can prove extremely painful to the instructor and can lessen the pedagogical and economic advantages which one is striving to attain.

PROGRAMMING ENVIRONMENT

FORTRAN was taught in the introductory course for several years. WATFOR was used in an attempt to simplify the material content (especially FORMAT statements) of the programming part of the course. Finally, BASIC was tried and has been quite successful. A batch compiler was written in PL/I at New Mexico State to process BASIC programs for the introductory course. It has several features that facilitate usage by a large lecture session (e.g., use of protected disk-based data files for reading by student programs). Unexpectedly the compiler has proven to be quite economical for our purposes. For example, approximately 500 students were assigned 8 programs each during the fall 1971 semester and the total time used on an IBM 360/50 was 1033 minutes or an average of about 15 seconds per student per program.

Even though the pedagogical advantages of BASIC are well known, some features are so ideal for the course that they deserve special mention. BASIC affords substantial power in the PRINT statement. With this one statement it is possible to have the student run several simple "programs" of the desk calculator variety. In the New Mexico State programming environment the student can run two-card programs: one card to identify him and his course and one card for his program. One has to have tried to teach programming to a very large class to appreciate the advantage of a one-card program; the advantage is substantial.

There are other pertinent pedagogical advantages to BASIC. For example, format specification by the picture technique in BASIC is far easier to teach than FORTRAN's FORMAT statement and automatic allocation of the first ten vector elements simplifies the teaching of subscripting.

QUANTITATIVE ASSESSMENTS

Several techniques have been used to assess the adequacy of teaching and of topic structure at New Mexico State. Student course and instructor evaluations are used for each semester and have engendered many course improvements. For example some of the currently used films have been selected on the basis of student reaction. This semester an adaptation of the Time Magazine/AFIPS questionaire, used in a recent national survey of the public's attitude toward computers, will be administered at the beginning and end of the semester. The goal is to obtain some quantitative measure of the effect the course has on student attitudes toward computers and computation.

V. CONCLUSIONS

The recommendations for introductory service offerings given in Section III have been found to be an effective way to meet the service needs at New Mexico State without excessive course proliferation. Even though disparities still exist within such consolidated course offerings, the resultant problems have been containable, if not solvable. The effectiveness of the present course structure has been substantiated by generally favorable student-completed course evaluations and thus it is felt that the structure is academically viable.

Finally the authors believe that the key introductory service courses are some of the most important courses taught by a department. They should receive the most broadly experienced and best teachers rather than being farmed out to new faculty or teaching assistants. With the substantial economic advantage afforded in such a consolidated environment, a department can ill afford to place other than their best qualified teachers in such courses.