COMPUTER SCIENCE FOR TEACHERS

Judith G. Malkin
The University of Texas at Austin

As computing systems in general, and timesharing systems in particular, are becoming less expensive and more reliable, a number of small colleges, junior colleges, and secondary schools are in a position to use computers in the classroom. Immediately this presents the problem of training and education, since many teachers may not have had the opportunity to learn about computers and computer programming when they were in school themselves, or if they had, a lapse of several years would have made much of their knowledge out of date. If a school is obtaining its own computer, the manufacturer may provide training in operation and programming, but this may have several drawbacks. There may be a cost associated with the training, the courses may be at a time prohibiting the attendance of the teacher, and, most importantly, the courses are probably not precisely what the teacher needs to use the computer in the classroom.

If the teacher is fortunate enough to be close to a university offering computer science courses, the same drawbacks mentioned above may apply. Many university computer science courses are either pure programming courses or highly theoretical courses. Courses in computer-assisted instruction (lesson design) also do not meet the teacher's needs.

With this in mind, and with a growing population of secondary school and small college users, the University of Texas Computer Science department offered a special course, specifically for teachers, in the 1971 summer session. Entitled "Techniques of Teaching Computer Science," which was later changed to "Computer Science for Teachers," the course was sponsored by the Southwest Region Educational Computer Network. The author, who is Network Coordinator, handled the administrative duties and taught the laboratory session. The lectures were given by the former coordinator of St. Stephen's Episcopal School, the first secondary school to join the Network. A description of the Network organization and operation is available (1). An initial outline of the course follows.

Week One:
- I. How the computer works
  - A. Software
  - B. Hardware
- II. Defining terms-Glossary
- III. History of computers
  - A. Brief account
  - B. Listing of references

Week Two:

Decisions and dichotomies in choosing a computer and setting up a program
- I. Computer Sciences vs. Programming
- II. Mainline vs. Adjunct
- III. Math and Sciences vs. Humanities
- IV. Rent vs. Buy
- V. Appreciation and Ethics vs. Use

Week Three:
- I. Resources
  - A. Libraries
  - B. Texts
  - C. Ads
  - D. Machines and games
  - E. Field trips

Week Four:

Computer-Assisted Instruction
- I. Survey of the state of CAI
- II. Taking sample lessons
- III. Principles to consider
- IV. Value and drawbacks

Week Five:

CLIC (Conversational Language for Instruction and Computing)
- I. Form
- II. Rules
- III. Uses
- IV. Practice

Week Six:

I. Identify interests and form interest groups

II. Plan and define projects

Week Seven:

I. Collect resources

II. Outline and begin projects

Week Eight:

Work on projects with consultation

Week Nine:

Evaluation of course, projects, and students' progress

The course offering was publicized in the Network Newsletter, and also by direct mailing to the superintendents of schools in Austin and several adjacent counties. We feel that direct mailing to the principals of the secondary schools in those counties would have been better publicity, as it seems to take time for information to filter down from the centralized district office into the individual schools.

The two students who were graduate Computer Science students learned of the course from the graduate adviser. Six of the students, who were teachers in the Austin public school system, received the publicity sent to the math and science supervisors. One student, who was also attending an NSF Summer Institute, heard of the course from someone at the institute orientation session.

The course listing appeared in the announcement of summer courses. It was listed as a graduate level course in Computer Sciences, but not cross listed in the education, science education, or math education departments. Although much of the material in the course was of an elementary nature, as can be seen in the outline, it was listed as graduate level in order to satisfy state education agency requirements on continuing education.

Backgrounds

Initially there were fourteen students, of whom three were auditors. Nine students of the eleven registered for credit finished the course (3 B's, 6 A's), and one of the auditors stayed the entire nine weeks, and completed all the assignments. Half of the students were 30 or under; only five of the fourteen had completed a master's degree or further study. Ten of the fourteen had either taken a course in which they learned to program or had worked with computers in the past--but the range of computer experience varied from a summer course ten years previous, to a candidate for a doctoral degree in Computer Science.

Eleven students did not feel that computers would take the place of teachers and other workers, while three were undecided, and twelve said that they would like to have a job (either in teaching or some other job) where they would use computers (one was undecided and one said "no").

The objectives of the students were fairly well divided into two major areas. About half of the students wanted to learn to become familiar with computers. They expected the course to provide them with an overview of computer programming, of how computers work, and to learn enough about computers and computer science to be able to teach the subject. The rest of the students were taking the course in order to get a background on computers in education: resources that would be available to them, techniques to use in the classroom in the fall. They all expected to be using the computers in their classroom, if not immediately following the summer course, then at some later date. One student was particularly interested in computer assisted instruction and wanted to learn to develop and use techniques in CAI for math and science courses. The students also indicated that they wanted to avoid tests and term papers, and they wanted to learn by doing things; by writing programs, by making observations. Several students specifically mentioned the avoidance of tests in a questionnaire that was distributed the first day of class, but we noticed about halfway through the course that many of them seemed to wish that we had given them tests, so that they had an idea of where they stood. This will be commented on more a little later.

The texts used for the course were as follows. (See also Appendix A):

1. Introduction to Computer Programming, by Crawford and Copp.
2. Computers and Teachers: A Beginning, Oregon State University Press.
3. Fortran Self Taught, by Bauer and Peluso.
4. The Bell System CARDIAC Kit.
5. An Atomic Energy Commission pamphlet entitled Computers.

The first three items were on sale in the University book store. The last two were obtained free of charge from Bell and the AEC and distributed to the class members. In addition, a number of articles, books, and pamphlets were available in the computer science library for our students. A complete list of these materials is available in

Appendix B. Also, several Computation Center publications on the use of the timesharing system, on the BASIC language, and on the CLIC course writing language were distributed to the students.

The approach used in teaching in both lectures and the laboratory sessions was an informal one. We explained that this was an experimental course and we would probably deviate from the syllabus as we went along, depending on the needs of the students taking the course. We explained our objectives, which were to make teachers familiar with computers, and to make them familiar with reference materials that were available. There were materials that could be used if they had access to a computer or to a compter terminal, and materials that could be used to teach about computers even if a computer or a terminal were not available.

We also wanted to teach enough of several languages so that they would, at least, be able to make use of programs that had already been written by other teachers. The three languages that we chose to teach were BASIC, Fortran, and CLIC. We chose BASIC because many educational materials from various sources are available in that language, and also because we believe it is a language that is most easily taught in the classroom. A student, after a few hours of formal instruction in the BASIC language, can write programs that run, and this we felt was very important, both in our teaching the teachers and in the teachers teaching their students. Fortran, of course, is a widely available language, and there are quite a few curriculum materials available in it. The third language, CLIC (Conversational Language for Instruction and Computing), is a course writing language that was developed at the University of Texas at Austin. It uses the operation codes similar to the Coursewriter III language, but since it is translated into Fortran, the CLIC programmer or course author can imbed Fortran statements within the CLIC code, and thus make use of any of the Fortran library subroutines that are available to him.

Approach

In teaching about computer hardware and software, we used many references that were available in the computer science library. Those students in our class who had had a great deal of computer experience, notably the two graduate students in computer science, were able to be of help in answering the other students' questions and in sparking a discussion about hardware, the different types of computers, and the different types of operations that computers perform.

We had the students assemble the Bell Telephone Laboratories CARDIAC computer kit, which is a cardboard computer with 10 different operations, and "run" programs on it. We felt that the CARDIAC system was a good teaching aid both where a computer was available and where it was not, and the reason that we used it in our class was because we felt the teachers might be interested in using it to teach their own students. Reaction to the CARDIAC kit varied. Several of the students did not like it. They felt it was much too complicated because it used a machine language. They found it easier to use BASIC or Fortran.

The laboratory assignments focused around using the TAURUS Timesharing System at the University of Texas. A beginner's manual on TAURUS was distributed to all the students, which contained both a sample program and step by step instructions on how to log in to the machine. We asked the students to read this manual, and then try to do the laboratory assignment, which was copying the sample program that was in the manual. By giving the assignment in this manner we thought we would very quickly learn whether or not the step by step instructions were as clear as we had thought. It turned out that when the students could not follow the directions, there was usually somebody else in the terminal room from whom they could get help. The first assignment, then, was to copy a prepared program and, using the manual as a model, to execute the program. Later laboratory assignments involved a discussion of the use of control cards or job control statements to perform certain tasks such as file manipulation, and other assignments involved copying programs out of one of the resource texts available, and getting it running on the system. The students were introduced as needed to the online text editor available in the system, and to the routines for permanent saving of programs and files. We didn't give any formal lectures on either the BASIC language or the Fortran language, since self-teaching manuals in both were available. We did, however, set aside several laboratory sessions just to answer questions on the languages and explain techniques such as loops, input/output statements, subroutines, etc.

Teaching the CLIC language was a little bit more difficult since the CLIC user's manual had not yet been published. We distributed some introductory material on the basic operation codes, and then one lecture session was turned over to a guest speaker, a girl who had graduated the month before from St. Stephen's Episcopal School. She had written a special project for both her math and German classes, which consisted of a set of four lessons in German grammar. These lessons were written using the CLIC language. The German teacher at St. Stephen's was so impressed with this student's effort that she has been using the set of lessons for the first year German students as a review. We felt that the experiences of this girl in designing and implementing the lessons would be of value to our students.

Later on in the course every student was given the assignment of writing one short program in BASIC, in Fortran, and in CLIC. The ideas for the programs did not have to be original; they could have been taken from one of the many reference sources that were available. The students had to turn in output from the teletype showing how they had created the program, edited it if necessary, and run it. Many of the students thus gained considerable experience in using the teletype and quickly realized that computers are not working 24 hours a day. We experienced a few days of system malfunctions in the summer, and it seemed that these days came just when we had given the students a large laboratory assignment. It was interesting to note which students were best able to cope with the frustrations of a computer being down.

We tried in the lectures to present techniques of teaching about computers by using games to illustrate certain concepts. There are several games that were available besides the CARDIAC kit. One of these is a "hands up, hands down" game, described in an issue of The Mathematics Teacher. It uses people to represent bits in a computer word, with a raised hand indicating a 1 and a lowered hand indicating a zero. In this way addition and subtraction can be demonstrated to the students, and the concept of complementary arithmetic can be gotten across much better than by lecturing about it and writing examples on the board. The other game was described in the March 1962 issue of Scientific American as a game-learning machine. With this game the students used match boxes and jelly beans of different colors to go through a series of moves, and the computer, which is made of the match boxes, is shown to remember the winning moves in the game.

We stressed at the very beginning of the course that grades were going to depend entirely on what the student expected from the course, what he put into it, and what he got out of it. It took a few weeks for some of the students to get use to this idea of grading. Several of them wanted tests, because they wanted to know where they stood in relation to other people, and where they stood in relation to what we expected of them. It was hard for some of these students to realize that we had no hard and fast expectations, that they were to set their own expectations, and then live up to them. We told them that we expected learning, but that what they learned and how they learned it was up to them. This was a novel concept to many of these teachers and they really didn't think that they could put this practice to work in their own classrooms. Their attitude seemed to be that this approach might work for graduate students, but that it wouldn't work with their students. We obviously tried to push our own theories of education as well as our ideas on teaching computer science.

Projects

The major objective of the course was to get the students to develop and implement some sort of project which they would use in their classes in the fall. We hoped that most of the students would have access to a terminal, but as it turned out only three of the students were in schools where terminals were available. Two of the summer students who had previously taught in Austin did not return to teaching for the current academic year. They were on Academic Year Institute fellowships and plan to resume teaching in September 1972. In addition, two other students began teaching in the Austin Public School system for the first time in September 1971, and although they requested placement in high schools where there were terminals, they were assigned to other schools. It is hoped that the Austin School Board will eventually equip all the high schools and junior high schools with terminals, but we don't know how long that is going to take.

The projects are briefly described in Appendix C. The scope and development of the projects varied greatly with the background of the teacher. Some had had considerable experience in computers, and so were able to undertake far more major projects than the ones who had not had any exposure to computers or programming before. The students submitted an idea for a project to us about midway through the course and in most instances we approved them as they were. With some we made suggestions for expanding, or in some cases for cutting down the scope of the project.

The last week of the class there were no lectures. The time was entirely devoted to working on projects and getting technical assistance from us, and just in general ironing out the bugs of the project. One project in particular, the Teacher's Outline for the teaching of the TAURUS timesharing system, has proven to be very popular and many of the teachers in the Network colleges and on the University of Texas campus are using it this year.

Conclusions and criticisms

The students had been asked to turn in, along with other written material, some comments about the course. On the last day of class they also discussed their reactions, and we were able to obtain several suggestions for improvements.

The biggest complaint about the course was that we were trying to teach too much, to too many people, in too short a period of time. In only one course, worth 3 hours of credit, we were trying to teach technical aspects, the programming, and the operation of the computer terminal, and education techniques. Many of the students were taking two additional courses during that nine week period, and it was difficult for them to devote the amount of time to our course that was desirable. So, most of the suggestions were to make it either a 6 or a 9 hour course, or to cut down on the amount of material covered. About halfway through last summer we realized that we were probably trying to accomplish too much, and so this summer we will be offering it as two separate courses: a three hour lecture in which we will concentrate on techniques and a three hour laboratory in which we will concentrate on languages and procedures. Any student who takes both of these courses this summer will be limited to taking only one other course, and we will strongly recommend that he not take any.

Other suggestions concerned items in the library that were being used for reference. Several students requested that copies of these be made available to take home, so that afternoons could have been spent at the terminal rather than in the library. We explained that most of these materials were not available in large amounts last summer, and will be this summer.

It was also felt that the reading lists were far too extensive and that we should have emphasized what was required reading and what was suggested reading for certain areas. Apparently many of the students, when told to read five articles, felt that they had to read and understand every single word in every single article. They weren't able to choose for themselves which articles could be skimmed or skipped.

Reorganization of the course was also suggested. A fair amount of beginner's material was handed out at various stages throughout the course, and most of the students felt that all of this material should be integrated and presented together at the beginning. The students also felt that there wasn't quite enough time for class discussion and suggested having one lecture period every week or one every two weeks devoted purely to questions and answers and discussions.

One thing that the students commented on was our having available a file of programs that high school students had written, and suggestions for programs to be assigned to students. At first they found it hard to believe that teenagers could think up such problems and then think up such sophisticated ways to solve them.

All in all, we believe that it was quite a successful experiment. We realize that the class was somewhat disorganized (and our students realized it too), because it was the first time that we had ever offered such a course. Even though we had tried to prepare thoroughly beforehand, we were still forced to change tactics as we went along, depending on the nature of the class. We have paid a great deal of attention to the specific criticisms that the students made, and we certainly hope that we will be able to improve on the course when we offer it again this summer. We would like to see this course become a regular part of the University of Texas Computer Science Department offering. In the future we will try to work more closely with the departments of math and science education at the University and perhaps put together a series of courses designed for teachers with the ultimate goal of a Master's in Computer Science for teachers.

APPENDIX A: Classroom Materials

1. CARDIAC Kit (cardboard "computer"). Available at no charge from Bell Telephone Laboratories.
2. Computers (pamphlet). Available at no charge from the Atomic Energy Commission, Oak Ridge, Tennessee.
3. Bauer and Peluso, Fortran Self Taught. Addison-Wesley.
4. Computers/Teachers: A Beginning. Oregon State University Press, Corvallis, Oregon.
5. Crawford and Copp, Introduction to Computer Programming. Houghton-Mifflin.

APPENDIX B: Reference Materials

1. Cooley, W. W. and Glaser, R., "The Computer and Individualized Instruction." Science. (166) 31 October 1969.
2. Atkinson, R. A. and Wilson, H. A., "Computer-Assisted Instruction." Science. (162) 4 October 1968.

3. General Information on CLIC (Pre-publication draft of User's Manual, The University of Texas Computation Center).

4. Douglas, A. S., "Living with Computers." Science J. October 1970.

5. Michie, D., "The Intelligent Machine." Science J. October 1970.

6. James, E. B., "Computers in Education." Science J. October 1970.

7. Joseph, E., "Towards a Fifth Generation." Science J. October 1970.

8. "Problems of Students Privacy Rights." Texas Outlook. July 1971.

9. "Computers and Privacy." Series of articles by A. R. Miller, Austin American-Statesman. June 28 - July 5, 1971.

10. Kemeny, J. G. and Kurtz, T. E., "Dartmouth Time Sharing." Science J. (162) 11 October 1968.

11. Allen, D. W., "The Seven Deadly Myths of Education." Psychology Today, March 1971.

12. You and the Computer. (pamphlet) Educational Publications, General Electric, Schenectady, New York 1965.

13. Stimulating the Development of Computer Based Educational Materials. The University of Texas Computation Center publication SRECN-2.

14. Bibliography prepared by Texas Education Agency Region 4 Educational Service Center, Austin, Texas.

15. Bell Telephone Labs. Record on computer speech. (c. 1963).

16. Cordell, C. M., "Computer in Your Life." Published by J. W. Walch, Portland, Maine, 1967.

17. Interrupt. Bulletin of Computer People for Peace. May 1971.

18. "This is How Computers Work." Changing Times, November 1968.

19. Introducing the Computer. (pamphlet) IBM Publication 520-1541.

20. Malkin, J. and McKay, L., Introducing the Computer into Small College and Secondary School Curriculum. Presented at the Houston Conference on Computer and System Science, Houston, Texas, April 1971.

21. "Can a Computer Think." Scientific American, June 1971.

22. "Can a Computer beat the horses." Look, 1 June 1971.

23. Project SOLO Newsletters. University of Pittsburgh Computer Center.

24. Crowley, T. H., Understanding Computers McGraw-Hill, 1967.

25. EDU. Digital Equipment Corporation Newsletter. Maynard, Massachusetts.

26. "The abacus." Scientific American (no date).

27. Nelson, T. H., "No More Teachers' Dirty Looks." Computer Decisions, September 1970.

28. Umans, S., "Computer in the Classroom." Parents Magazine (no date).

29. "How to build a game learning machine." Scientific American, March 1962.

30. BASIC Mini Manual. The University of Texas Computation Center publication TPB-124.

31. Project CLUE Glossary of Computer Terms (pre-publication draft).

32. Dorn, W. S., "Computer-extended instruction, an example." The Mathematics Teacher, February 1970.

33. Beckmann, M. W., "Teaching the Low Achiever in Mathematics." The Mathematics

Teacher, October 1969.

34. Problems for Computer Application, Southern Region Education Board, Atlanta, Georgia.

35. "Computer for School Mathematics." The Mathematics Teacher, May 1965.

36. Kiewit Computation Center outline guides. Dartmouth College, Hanover, New Hampshire.

37. Resource sets in science, mathematics, and business. Oregon State University Press.

APPENDIX C: Class Projects

Note: The scope of the projects varied greatly in accordance with the backgrounds and objectives of the students.

1. Two CLIC lessons on the use of permanent file storage on the timesharing system.
2. A Fortran program for computing and printing the resultant of two or more vectors in a plane. For use in secondary school physics classes.
3. A BASIC program to resolve a combination Gaussian-Lorentzian expression for observance.
4. Three simulation studies programmed in Fortran: Arrivals and departures of aircraft at a one-runway airport, motion of a rocket, and a coin-tossing experiment.
5. A set of BASIC programs and exercises on forces, to be used in a high school physics course. Includes a pretest.
6. A teacher's outline to the use of the timesharing system, including simple exercises using control statements.
7. A programmed text on genetics, for a junior high school biology class. Will be adapted for computer use when the school gets a terminal.
8. Outlines for teaching Fortran and assembly language in secondary school.
9. Outlines for using the computer in a mathematics course.

REFERENCES

1. Southwest Region Educational Computer Network: The Second Year, A report to the National Science Foundation. The University of Texas Computation Center Publication SRECN-3, July 1971.