Everything You Always Wanted to Know About Computers . . . You Can Teach Yourself

Jesse M. Heines

It Types By Itself!

The chilling fear that computers will dehumanize education is slowly taking its rightful place among the archives of unfounded educational beliefs. This is due to several factors, but the most important by far is simply the growing existence of the computer in instructional settings. You can talk all you want to about how exciting computers can be, but there is no equal to sitting down at a teletypewriter and conversing with a computer for understanding what these machines are really all about.

High school students are notorious for their incredible love of computers, even to the point of coming back to school at night to work with the interactive number crunchers. Yet all this *student* excitement sometimes scares off more *teachers* than it attracts. Teachers have more hang-ups than students about breaking things, so they usually like to understand the machines that they are working with before experimenting freely. In addition, few of us like to make mistakes when others are breathing down our necks waiting to take over our places. The problem is similar to learning how to dance: you certainly don't want to learn in front of all your friends, but somehow you can't really learn at home with just your record player, either.

Rubs and Ruses

Unfortunately, there are very few places that teachers can go to learn about computers. College courses are usually theoretical or demonstrate things on large computers that are quite outside the realm of the minis which serve our nation's secondary school classrooms. Introductory courses offered by computer companies tend to deal with computer architecture or assembly language, topics which simply fog the issue for teachers who want to know how to use the computer in their own courses.

If you go to your local student computer freaques for assistance, they will probably perpetrate the rub that the workings of a computer are highly complicated. They're not lying, you know, but their rub is really a ruse: if *you* don't use the computer, it leaves more time for *them*.

What your local student computer freaques probably *won't* tell you is that *you really don't have to know anything about the internal workings of a computer in order to use one*. Think a minute. Do you have to know how the shutter works in a movie projector to view a film? Do you have to know how light is refracted to use a microscope? Do you have to know how an internal combustion engine works to drive a car? Of course not. All you have to know is how to *operate* a machine properly and you will be in control.

Teach Yourself

There is only one recourse: teach yourself. (After all, where did the freaques learn?) Teaching yourself about computers is probably far easier than you think it is. The following sections suggest people and organizations to whom you can write for information on using computers and outline a self-instructional scheme that you can carry out simply through reading their materials and experimenting with the machine in your school.

As a teacher, you will want to be familiar with four topics in instructional computing to use the computer effectively in your classroom. These are: program development, courseware transportability, project planning and resource identification. Each of these topics is explained below with suggestions for obtaining mastery of the skills involved.

Program Development

Computers are controlled by computer programs. In most problem-solving situations in which the computer is used, people either write programs to help them solve problems or modify programs written by others to match their specific needs. While it is not necessary to be an expert programmer to supervise student computer projects, it is necessary to be familiar with some programming concepts and have some first-hand programming experience to help students in using the computer effectively.

All computer vendors sell manuals which explain the complete operation of their machines. More often than not, the manuals which complement instructional computers are written by teachers rather than engineers, so they are quite readable by even the most naive beginner. These manuals often contain not only directions .for running the computer, but also copious examples

Jesse M. Heines is a training course developer, Educational Services, Digital Equipment Corporation, Maynard, Massachusetts.

of programs that might be directly usable in your classroom. Manuals are usually very reasonably priced, as most vendors rely on the sale and service of the actual computer itself to cover their costs.

The first step, then, is to order yourself a computer manual. To do this, go see your system manager (the keeper of the computer) and find out the address to which you should write, the exact name and number of the manual written for your computer and/or the model name and number of the machine itself. This information will greatly expedite your order.

While you are there, chat a while with your system manager. He/she is probably an old computer freaque himself/herself, and a lot more willing to help you than the student freaques. You see, your system manager would probably *love* to see you use the computer as much as possible so that he or she has an ally when it comes time to try to convince the school board that you need more terminals on your machine. More altruistically, he or she can probably suggest several score of interesting projects that would be fun to do on the computer, but for which he or she has not been able to find the time.

Once you receive your manual, return again to your system manager. Ask him or her when you can come in to work on one of the terminals *privately*. Believe it or not, if you read the manual and spend as little as three to five hours at a computer terminal experimenting with the various techniques discussed, you will have an excellent grasp of what computer programming is really all about. The best way to learn, however, is with *no outside assistance*. This will force you to use the reference sections of your manual and familiarize yourself with the error messages. The computer itself is very forgiving, as almost *no error is totally unrecoverable*. The only really unrecoverable error is the deletion of a needed file, but this can be avoided if you keep duplicate copies of those programs that you are actively editing and modifying.

One further piece of advice while you are writing programs to try out the commands in your manual: begin as early as possible to write a program to help you solve one small aspect of a clearly defined problem or task. As you learn more commands, expand this program to make it perform more functions, run faster and require smaller areas of your computer's memory. This way, you will have a highly usable product by the time that you have learned all the commands in your manual.

Courseware Transportability

The words "hardware" and "software" are commonly used computer terms and refer, respectively, to the actual machinery and operating system (or control programs) that make a computer capable of doing anything at all. Teachers must also be concerned, however, with the instructional *content* of any programs that they give their students to run. This content is known as "courseware." One can demonstrate the relationship of these three "wares" by using the common text-book as an analogy. The actual cover and pages of a book are its hardware. The language in which the book is written is its software, and the concepts and ideas that it conveys are its courseware. Thus, a text printed on microfilm is useless unless you have a microfilm reader (the right hardware), a book written in Russian is useless unless you can read that language (the right software), and the best treatise on plant morphology is useless if you are studying Newtonian physics (the right courseware).

Fortunately, most of the computers used in educational institutions have certain similarities that make their programs highly transportable. Courseware is said to be transportable if a computer program written on the computer at *my* school will run on the computer at *your* school with only minor modifications and supply you with a valuable teaching tool. By keeping a few points in mind, you can easily optimize courseware transportability without sacrificing one iota of instructional quality.

Hardwarewise, all instructional computers use some form of teletypewriter to communicate with their users. These may print on paper or TV-like screens, and printing speeds may vary greatly, but all are interactive, allowing their users to conduct a pseudo-dialogue with the computer. Therefore, your courseware should take advantage of this characteristic and involve as much interaction as possible. If you want to present a great deal of textual material, *write a handbook* to accompany your computer program. Nothing is quite as annoying when you go to use the computer as to find that a terminal is busy churning out miles of printout while its user has gone to attend to other business.

Softwarewise, stick to the BASIC language. Some schools are getting into FORTRAN at the secondary level, but you will find BASIC much more widely implemented in instructional settings. You should definitely read two excellent articles which appeared in the *Bulletin* of the Association for Computing Machinery Special Interest Group for Computer Uses in Education. These are "Universal BASIC: A Way to Reduce Conversion Costs" by Ronald W. Confer* and "Interdialect Translatability of the BASIC Language" by Gerald L. Isaacs.** (The address of the ACM SIGCUE publisher appears at the end of this article.) These two articles provide excellent guidelines for optimizing courseware transportability. As you write programs and supervise those written by your students, stick to the conventions outlined by Confer and include a generous number of REMARK statements (comments which are ignored by the computer) to document the programs for other users. Your programs, however, should never *branch* to REMARK statements, so that these statements may be deleted by others without affecting the integrity of your program if their computers have smaller memories than yours.

Talk to your system manager about setting up a program library in your school. Most of the computer programs that you would write for use in your classroom if you had the time have already been written by other teachers. (Many should be available from your computer vendor; ask your sales representative.) Each time a complete program is written, it should be *documented* with its name, a statement of its purpose, the prerequisite skills and materials needed to use the program (if any), references (if any), a program listing, a sample printout and recommendations for future expansion. A good test of a well-documented program is simply to give it to other teachers and ask them to use it with no additional help from you. If they can, it is well-documented and transportable.

Project Planning

Next, you will want to become more experienced in guiding student programming efforts and more familiar with the exact constraints imposed by the particular computer installation at your school. Once you have some programming experience, define a new problem or task and write a program from scratch to help accomplish your new goal. This will aid you in realizing just how long it really takes to write a program, provide a basis for estimating the scope of any projects that your students may wish to undertake, and give you a realistic idea of the kind of work that you can expect from your students within given time frames.

With these ideas in mind, meet with your colleagues and system manager to plan a large-scale project that will require several classes to interact with the computer. Among other ideas, you might plan to write a single program with which many students will interact, conduct a contest in which many students will try to program the computer for a specific task, or define a problem and assign groups of students to solve its various aspects with the computer. Such a project will most likely push your computer to its full limit of service and provide an excellent understanding of its capabilities and limitations. Pare down your original ideas and plan and execute the large-scale project with

*ACM SIGCUE Bulletin *8* (2), April 1974, 3-9.

**ACM SIGCUE Bulletin *8* (4), October 1974, 11-22.

full orchestration. One effort like this and you will know as much about your computer's usage as anyone in your educational community, and you will have the raw data for an interesting paper to submit to an educational journal.

Resource Identification

The final step in teaching yourself about computers is to know where to find further resources. Several suggestions are listed below.

First, write to the Educational Products Groups at various computer companies (some names and addresses appear at the end of this article). They will be more than happy to send you reams of sales brochures, and you will find these easy to read. The vendor who sold you your machine will be able to put you in touch with other people and organizations who can supply you with courseware developed specifically for your computer.

Next, join a couple of professional societies that deal with instructional computer uses and/or subscribe to professional magazines in the field (see the partial lists at the end of this article). As a teacher, all the expenses that you incur through these professional memberships should be tax deductible! (Consult your tax return preparer for specific details and advice.)

Through these organizations, their journals and other professional magazines, you will learn the names of other teachers who use the computer in their classrooms and who are usually very happy to have their materials utilized by others. They provide the third resource for information on instructional computing and one of the best sources of transportable courseware.

Be Your Own Freaque

With little more than average enthusiasm, you can easily teach yourself everything you always wanted to know about computers within a single year's time. Who knows? You may even show the freaques a thing or two! Just be careful that they don't put you in charge of the computer itself, or you will have to be polite and let others use it, too. □

Partial List of Resources
(in alphabetical order)

Computer Companies (address inquiries to the Educational Products Groups)

Digital Equipment Corporation
146 Main Street
Maynard, Massachusetts 01754

Hewlett-Packard, Inc.
11000 Wolfe Road
Cupertino, California 95014

Wang Laboratories, Inc.
836 North Street
Tewksbury, Massachusetts 01876

Magazines

Creative Computing
P.O. Box 789-M
Morristown, N.J. 07960
$8 for 6 issues per year.

EDU
Educational Products Group
Digital Equipment Corporation
146 Main Street
Maynard, Massachusetts 01754
Free, 4 issues per year.

Educational Technology
Educational Technology Publications
140 Sylvan Avenue
Englewood Cliffs, New Jersey 07632
$25 for 12 issues per year.

People's Computer Company
P.O. Box 310
Menlo Park, California 94025
Free, 12 issues per year.

Professional Societies

ACM SIGCUE (Association for Computing Machinery Special Interest Group for Computer Uses in Education)
P.O. Box 12105
Church Street Station
New York, N.Y. 10249
$6 per year membership fee. Publishers of the *ACM SIGCUE Bulletin* (quarterly), subscription included with membership fee. Conventions held in conjunction with other ACM activities.

ADCIS (Association for the Development of Computer-Based Instructional Systems)
P.O. Box 70189
Los Gatos, California 90070
$10 per year membership fee. Publishers of the *ADCIS News* (bimonthly) and the *Journal of Computer-Based Instruction* (quarterly), subscriptions included with membership fee. Two conventions per year.

AEDS (Association for Educational Data Systems)
1201 16th Street, N.W.
Washington, D.C. 20036
$20 per year membership fee. Publishers of the *AEDS Monitor* (bimonthly), subscription included with membership fee. One convention per year.