DOCUMENT RESUME

ED 267 412 CS 209 594

AUTHOR Collins, Terence; Price, Lynda
TITLE A Guide to Selecting Word-Processing Software for Learning Disabled College Writers. Working Paper.
INSTITUTION Minnesota Univ., Minneapolis. General Coll.
SPONS AGENCY Department of Education, Washington, DC.
PUB DATE Feb 86
NOTE 17p.
AVAILABLE FROM Learning Disabled Writers Project, University of Minnesota, General College, 106 Nicholson Hall, 216 Pillsbury Dr., S.E., Minneapolis, MN 55455 ($1.00, prepaid, checks only).
PUB TYPE Guides - Non-Classroom Use (055)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Adults; *Computer Assisted Instruction; Computer Software; *Courseware; Higher Education; *Learning Disabilities; *Microcomputers; *Research Projects; *Word Processing; Writing Instruction; Writing Skills
IDENTIFIERS Learning Disabled College Writers Project

ABSTRACT

Since word-processing applications of microcomputers show a great deal of promise in helping learning disabled adults overcome writing problems, care should be given in selecting the software that they use. In considering what shared needs must be addressed when selecting word-processing software, it is necessary to isolate some of these needs, including the need for self-expression; a format for self-expression that meets standard "receiver needs"; a process of writing that is flexible enough to bypass fine motor, eye-hand coordination, or spatial orientation problems; and a format of expression that has a consistent structure to accommodate their problems with transfer or synthesis of specific information. Five selection criteria for word-processing software should be considered: (1) visable program logic, (2) clarity of on-screen format, (3) the manufacturer's documentation and tutorial, (4) on-screen working features, and (5) multisensory approaches. (A screening checklist that focuses on desirable features intended to accommodate the needs of the learning disabled writer is included.) (EL)

A Guide to Selecting Word-Processing Software for Learning Disabled College Writers

A Working Paper of the Learning Disabled College Writers Project of the University of Minnesota - General College, February, 1986

Terence Collins,
Director

Lynda Price,
Learning Disabilities Specialist

LD College Writers Project
University of Minnesota - General College
106 Nicholso' Hall
216 Pillsbury Drive SE
Minneapolis, Minnesota 55455

Research supported in part by a grant from the
U S. Department of Education

Terence Collins
Lynda Price
University of Minnesota-General College
106 Nicholson Hall
216 Pillsbury Drive S.E.
Minneapolis, MN [illegible]55

A Guide to Selecting Word-Processing Software for Learning Disabled College Writers

A Working Paper of the Learning Disabled College Writers Project of the University of Minnesota-General College, January, 1986

Word-processing applications of micro-computers show a great deal of promise in helping learning disabled adults overcome the tremendous problems with writing that until now have prevented them from succeeding in academic and career areas of their lives. Any number of anecdotal reports in articles and conference papers anticipate or confirm the preliminary findings of our research in the University of Minnesota Demonstration Project for Learning Disabled Writers: carefully presented, word-processing has positive impacts on both the attitudes and performance of learning disabled college writers. While the controlled studies that would give experimentally-validated credence to these frequent but scattered reports remain to be completed and published, the testimonials make it likely that more and more of us who teach writing to learning disabled adults will turn to word-processing for help.

An essential and early part of the care that must be put into introducing word-processing to l.d. writers is software selection, and the writing teacher or l.d. specialist who looks to word-processing for help will find that the task of software selection is not easy. Even with the help provided by reviews in journals, the individual professional in the field is faced with an isolated software purchase decision reflecting the given institutional budget, hardware availability, shared-use or multiple-use constrictions placed upon the available machines, and degree of specialized support provided. And the selection made is one that we live with for a long, long time, given the cost of a good word-processing application product. No matter how sophisticated or new or shiny the hardware, if the software isn't chosen critically with the special needs of the l.d. user in mind, the potential benefits of word-processing as a writing tool are diminished or negated. Without software that meets the l.d. student's disability halfway, word-processing becomes just an expensive high tech opportunity for failure for the student and wasted resources for the institution.

The Typical Configuration of Needs for L.D. Writing Students

Learning disabled people bring to any situation individual patterns of strength and weakness, of course. But based on the familiar and widely shared processing styles of l.d. writers, it is possible to isolate some shared needs among a variety of l.d. writers and to posit them as a starting point for beginning to consider what shared needs must be addressed in word-processing software application programs. Naturally, not all of these needs are unique to to the l.d. writer: teachers who work with college and adult students who fall unr the broad heading of "basic writers" will be on familiar ground here.

Learning disabled writers, then, generally need:

1. a format for self-expression that is less cumbersome and less time-consuming than either cursive writing or typing;
2. a process of self-expression that is itself "invisible" -- that is, a process that doesn't become a distraction or a handicap itself;
3. a format for self-expression that meets standard "receiver needs" -- that is easily understood by the audience of the writing;
4. a proces of self-expression that taps the creativity and "global" learning style frequently encountered among l.d. students;
5. a tool for self-expression that is not unique to the accommodating setting, one that can be adapted broadly and readily for academic, personal, and occupational success beyond the context of our helping intervention;
6. a process of writing that is flexible enough to bypass their fine motor, eye-hand coordination, or spatial orientation problems;
7. and perhaps most importantly, a format of expression that has a consistent structure to accommodate their problems with transfer or synthesis of specific information.

In reality, no one piece of software can meet all of these needs, and the selection process is really a matter of balancing strengths and deficiencies among a variety of products. For example, the workplace applicability of Wordstar is certainly something to be considered, as is the ease of use in a product like Bank Street Writer or the Milliken word processor. Individual contexts will dictate whether Wordstar 's complexity and poorly written documentaion make it worth the vocational carry-over benefits or whether Bank Street 's simplicity and nearly universal success balance the complete lack of vocational application.

Selection Criteria for Word-Processing Software

Meeting or even approaching the needs of the l.d. writer is, obviously, a matter of degree -- no one has yet discovered word-processing to be a panacea. The degree to which a piece of software meets these needs, though, can be assessed. In general, such software is marked by common characteristics. These characteristics are explained here and are then revised into an evaluation checklist that we have used successfully in the preliminary screening work for the Learning Disabled Writers Demonstration Project at the University of Minnescta.

First, visible program logic: On-screen, does the program have a linear, step-by-step structure that is progressive, straightforward, and clear? Does the program move the student from mode to mode and task to task with as few disruptions . distractions as possible?

One symptom of a program which is succesful under this criterion is that the program require simple, familiar keyboard commands to enact the different modes of writing, editing, saving, printing and such. (The new "mouse" technology, requiring pointing rather than keybaording for these shifts, can be similarly evaluated.) For instance, the popular and widely adopted Applewriter II, an intermediately powerful program, is attractive under this criterion because it relies on control-key commands that are intelligently mnemonic in concept. To move to the beginning of a document, the writer simply presses the control key and the B (for "beginning") key; to move to the end of te document, one simply presses the control and the E keys; to change to the printing mode, the writer hits control and P keys; to save, control and S; to load a file, the control and L keys; and so on. This simplicity of design, using a constant -- the control key -- and a logical mnemonic auxiliary like P for print, gives the l.d. user a running start at mastering a fairly powerful program.

Another form of this visible program logic is found in products like Homeword. In this type of program, the choices are displayed througn pictures or icons which describe the mode to be chosen or the command to be enacted. In Homeword, one moves the lighted bar on the screen, using the "space," to select, for example, a file cabinet (to which files are saved or from which they are loaded) or a typewriter (to print) or a desk (for writing or editing).

Less desireable for l.d. users are those programs -- many of them excellent for general use, by the way -- which use less logical, less easily recognized commands and which require more than a single step to enact a command. As a rule of thumb, it seems like the more powerful the program, the less likely it is to meet the criterion of visible program logic. Not surprisingly, many of the more sophisticated programs for IBM PC compatibility fall into this category. A typical example might be found in WordPerfect, a program used widely by non-l.d. writers in business. In Wordperfect, to change a letter from single to double-spacing, the writer must press eight different keys, including three different function keys. These keys are labeled numerically (as in F2, F3 etc.), so the writer must know the documentation or use a template to recall the commands. There are so many diverse key commands, that WordPerfect comes with a fourteen page "quick reference" guide.

The second characteristic is clarity of on-screen format: This is closely related to the previous criterion, visible screen logic, but has more to do with how the user is presented with choices than with how those choices are enacted. Clarity of presentation of choices is as important as ease of enactment of those choices for the l.d. writer, especially for those who bring histories of reading comprehension deficits to the writing situation.

Several types of menu selection are available. As mentioned above, Homeword and Milliken display their menus through icons or icon-word combinations and are enacted by moving the spacebar, and our experience is that l.d. users welcome this escape from the highly verbal format of word-processing, with its barrage of new terms and technical language.

More powerful products increasingly rely on simple keystroke commands for menu displays, such as Applewriter II, Microsoft Word, and Volkswriter. For these, the question that needs to be asked is whether, once it is displayed via the "help" or "panic" keystroke, the menu is usable. Most students find plain-English menu choices helpful, such as "Move some text" as opposed to "Block Insertions" or "Delete text and save" as opposed to "Delete to buffer." As is the case so often in software use, the menu can either draw on the student's previous history with language and thereby promote confidence and success or it can use tech-babble and promote panic an failure.

Another consideration is the amount of information or number of choices provided by the menu at any one time. Our experience is that the l.d user is easily distracted by whatever stimulus is present, and if the menu is providing too many or irrelevant choices, these extra pieces of information -- presented all at once for the sake of efficiency, no doubt, for non-l.d. users -- become distractors and slow down the writing process. One of the commonly cited drawbacks among our students using *Macwrite* is that too many choices are on the screen at once. The student is asked to consider such things as typeface or mode while writing is under way. Again, the popularity of the Macintosh and *Macwrite* among even very young or very scared non-l.d. users suggests that this poses no problems for mainstream users and is even seen as a plus -- but it poses some problems for our l.d. writers. It's a good example of how even the best designed materials might create problems for this population.

Third, the manufacturer's *documentation and tutorial.* Most vendors provide instruction materials with all software Some provide on-disk tutorials to supplement or introduce those instructions. Still others provide tape-recorded instructional sessions to get the first-time user familiar with the program's basic operations.

In no other area are the needs of l.d. users so difficult to accommodate. Most documentation is just plain badly written. Word-processing manuals that leave Ph.D. colleagues frustrated and self-doubting are not likely to fare much better in guiding l.d. student writers. In deciding whether to adopt a word-processing application product, a question that must be considered is how much rewriting of documentation or writing of supplemental instructions will be required if the l.d. students are to become independent users of the software. Here again, the question is not likely to be whether you will need to write your own instruction materials, but how much time and effort such rewriting or supplemental writing will take to make the product usable.

The first and most obvious desireable characteristic in the manufacturer's documentation is that the language used in the tutorial be as free of needless jargon and technical language as possible. There are some specialized terms, of course, which must be used in writing about computer programs, and all software will use them -- after all, it's hard to talk about a cursor without using the word "cursor," since the alternative (perhaps "blinking green box"?) is more cumbersome. But such terms are few and must be clearly defined and visually illustrated in the documentation. It doesn't take long to get a feel for whether the documentation has been written with as few distracting flights of vocabulary as possible or whether the key terms are presented clearly.

Less obvious issues are the size of bites of information that the reader is expected to take in in one chapter and the logic of such chunks of information. The best documentaton, like that for the simple Bank Street Writer, has short, discrete sections that present only a few closely related operations. Perhaps because it was written for middle-school users rather than for business applications, it is mindful of limited attention and comprehension of users. Related to this consideration is whether the documentation and tutorial allow for different levels of sophistication of use for the program, or whether it assumes that all users will and must learn all applications. Good documentation and tutorials are sectioned so that a beginner can learn a few basic operations and set the material aside until he or she outgrows these basic uses. Weak documentation takes the user through all of the options in a given mode. And, unforunately, it does so without recognizing that some users, like beginning college freshmen, l.d. or not, have no use for many features such as embedded footnoting, mailing lists, form letters, or integration of graphics and spreadsheets.

It is likewise important to assess how much familiarity with the hardware is assumed by a given product's documentation or tutorial. PC Write, for instance, is a very good, cheap, frequently updated program for IBM PC compatibles. Its low cost, free updated versions, lack of copy-protection, and friendly support make it an attractive program, at least at first glance. But l.d. users would find that it assumes a fairly high degree of familiarity with the IBM PC, a machine that is somewhat intimidating to beginners. Moreover, the documentation suffers from some needlessly complex sections. On the other hand, some of the simpler products written for Apple IIe use, like the Bank Street Writer and Milliken word processing program, provide enough information about the Apple so that a complete novice to the machine can be writing successfully in the first session.

It bears repeating to note that our experience with l.d. college writers in our Demonstration Project suggests that if the documentation isn't clear, the l.d. specialist or writing teacher faces the task of rewriting or supplementing the stock materials. Most l.d. students already fear writing and distrust their own experience as writers to such a degree that their first encounter with the instructions for the new tool of word-processing can set the tone for their subsequent use of the technology.

The fourth category of concern, on-screen working features, is a general one. Besides those considerations of menu and on-screen movement covered above, it is important to work with the word-processing product being considered for adoption to note some important points. Does the product have a preview format option that lets the writer see how the text entered on screen will actually look on the printed page? Many l.d. students have trouble imagining a screenful of shadows, sometimes with embedded commands, as a printed paper, so this feature is very helpful for editing. Unfortunately, only the more powerful programs have this option and exact a heavy price from the user in what he or she has to give up in terms of clarity or ease of operation. Some program have 40 or 80 column display options built into them, although most, other than the simplest like Milliken, present in an 80 column format automatically. Shifting back and forth between 40 and 80 column display is a nice feature for some writers: 80 columns let them see more of their text; 40 columns let the see it "better." Individual students will have individual preferences, and it is convenient to accommodate both when possible.

There is a decidedly undesireable feature of presentation in some older word-processing packages for use on micros and found on some dedicated word-processors as well. This is the program's presentation of only a part of a line of text at a give time. All word-processing packages "scroll up" text when the screen is full. These dated programs and processors work similarly to the left and right within a given line, allowing the student to see the word or phrase in progress, but moving the line of text out of the student's vision they undermine coherence and continuity. The older Magic Window word-processing program was thus flawed, and caused no end of aggravation to our l.d. students who used it. Similar concerns might be raised by programs (or old hardware) that don't allow for upper and lower case displays. Newer software, including new editions of some old products, have addressed these concerns nicely, but persons inheriting equipment may have some cause for concern.

The fifth characteristic desireable in word-processing is use of multi-sensory approaches (perhaps in the tradition of the Orton-Gillingham method) in both orientation to and use of the program. An obvious plus, alluded to above, is the presence of a taped tutorial with the program that lets the novice l.d. user focus on the screen while being instructed in the auditory mode, rather than shifting from reading printed documentation back to the screen and keyboard. Many students find taped instructons a reinforcing tool that allows them to "overlearn" the material through a second sensory mode once they have worked through printed instruction (a handy feature of such programs as Homeword and, depending on which version you get, Macwrite). This appears to be especially beneficial for l.d. students who are currently utilizing taped texts for ther courses.

A desireable characteristic in a program for l.d. writers might be the "whistles and bells" that are featured in some programs. Here, when a terminal command, such as dumping some text or erasing a file is about to tbe executed, a bell or beep will be emitted while the screen flashes a warning that the next move is an important one. This feature serves to reinforce visually-given information with simultaneous auditory cues.

In this vein, some software calls for modifying the keyboard with color codes and icons that make it more accessible than does the gun-metal gray of most stock keyboards. Visual highlighting or separating by color certain keys or or keyboard features make special function keys leap out at the user. It is, of course, possible to design your own keyboard modifications and colorings to enhance whatever software you settle on. This may be necessary for l.d. students who currently use visual clues to decode texts or clarify written materials. Other products similarly use varied colors on the screen, separating instructions from text and the like, thu· helping the l.d. user focus on one subset of the display at a time. One potential drawback of this is that color monitors tend not to give as good resoluton for 80 column displays as do monochrome screens.

Perhaps the most exciting possibility for mixed media auxiliary tools for l.d. writers using word-processing is the voice synthesizer which "reads" in poor to fair auditory form the student's text. It is attractive for two reasons. First, it provides a 'stupid" reader -- one who gives back only what the student has entered, one who reads mistakes as they are given, not as the student would have had them be -- allowing the l.d. student who cannot visually decode his or her own editing oversights to hear the garbled speech of the machine stumble over them, flagging them for revision or correction. Furthermore, it provides simultaneous verbal stimulation through both he visual and auditory channels. We hope to test this feature in the University of Minnesota Learning Disabled Writers Project, using Macintosh's Smoothtalk and Echo speech synthesizers.

Screening Checklist

Each program has its own goals and students, and so it makes sense for each program to develop from its context a screening checklist. The virtue of such a checklist is that it focusses the attention of various reviewers on those features in a word-processing product that have been isolated beforehand as being desireable. Since trial screening is sometimes done on the crowded salesfloor of a computer store or at home away from the chance to interact informally with colleagues, the checklist helps to insure that all features relevant to the decision are addressed by all reviewers, regardless of the conditions under which the product is reviewed.

[insert checklist about here]

Our checklist reflects our sense of the needs of the l d. college writer outlined above and our sense of what the desireable features in a product to be used by l.d. writers would be. We have limited our review to products which operate on Apple IIe, IBM PC compatible, and Macintosh hardware, and our criteria may reflect te advantages and limits of those widely adopted machines. Users of Commadore, Radio Shack and other machines will find our general checklist a good starting point, however.

Suggestions for Criteria to Evaluate New Word Processing Software for Purchase

yes	no	
		1) Does the software have a step-by-step structure that is: systematic? progressive? clear?
		2) Does the software have a tutorial that is: easy? clear? comprehensive?
		3) Does the software have a simple menu format? Are the menus as discrete as possible with little flipping between menus?
		4) Is the screen presentation of information clear? Does it use a linear format? Does it use a lot of visual stimuli to present information?
		5) Is there as little technical or obscure vocabulary as possible?
		6) Does it use as few disks as possible?
		7) Does it use a multi-sensory approach if possible? (example: color-coded keys, warning beeps, etc.)
		8) Does it teach too many unused skills at once? Does the tutorial define foundation skills or give examples?
		9) Do the skills provided in the software justify its cost?
		10) Are extra peripherals or hardware needed?
		11) Do the plusses justify the expense of this purchase?
		12) Is much previous computer experience necessary to run the software? Can the student receive it at your facility?
		13) Can basic college writing tasks be completed using the product?
		14) Is it practical for your students?

LAP 8/85

13A

We use the checklist as a starting point for discussion, not as a final judge. Not even harried academics should be arrogant enough to assume that a brief checklist can do justice to a product that teams of professionals have worked months and years to perfect. As a focus for discussion, though, it makes the task more focussed and economical. Moreover, it helps explain decisions that lead to fairly large expenditures of limited money (and if you are outfitting ten machines with $79.00 Bank Street Writer instead of $400.00 Wordstar programs -- or vice-versa -- you may well need to jusify your choices), and it helps to insure that you will be happy with your decision for the long life of the program.