RESEARCH IN

WORD PROCESSING

NEWSLETTER

South Dakota School of Mines and Technology
Rapid City, SD 57701
Volume 5  Number 3

(605) 394-2481
March 1987

Software Review: PC-WRITE 2.7
John Ogasapian

Bibliography Update
Bradford A. Morgan

Evaluating Student Papers with a Word Processor: A Progress Report
William S. McCleary

Computerized Evaluations for Writing
see page 9

2

6

9
Software Review: PC-WRITE 2.7

John Ogasapian

Program: PC-WRITE 2.7

Available From: Quicksoft
219 First N. #224
Seattle, WA 98109
(206) 282-0452

Price: $16 for the two disks, applications and utilities, and "quick guide" (on disk); $89 for full registration, including manual and support.

Requires: IBM PC/XT/AT or compatible, one double-sided drive, 256K RAM (320K for spell check).

Version 2.7 of Bob Wallace’s PC-WRITE is now available, the program’s fourteenth upgrade in a little over three years. Version 2.55 and its immediate predecessors, regarded as a sort of standard for MS-DOS middle-level word processor, have given way to a package at least one major periodical reviewer has bracketed with the top programs: Word Perfect, Microsoft Word, and XyWrite.¹ No doubt about it; PC-WRITE 2.7’s editor is as fast as any and faster than most, versatile, easily modifiable, and intuitive. But its potential is held back by a separate dot-line formatter/printer and its speed underutilized because of the 60K limit on the size of any single file. Of this more anon.

Three years is an eon in computer time-scale, and over that period PC-WRITE has attracted a large following. It is widely used, if reports are accurate, among students and faculty on a number of campuses for varying levels of serious writing. According to Quicksoft’s own literature, the program also is employed officially by a range of firms. Among education institutions are Harvard University’s Business School and the U.S. Naval Academy: on reflection, highly portentous for PC-WRITE’s future market position, given the monogamous attachment often formed to the word processor first learned on one hand, and the influence on purchasing decisions a decade or two hence being incubated in the classrooms of those two institutions. Among business users are Walt Disney World, Eastman Kodak, the two chemical giants Dow and Monsanto, Arco, and even IBM.

Some 40,000 copies of PC-WRITE have been distributed over the years by Wallace’s firm, and countless more have been copied and shared by users’ groups, clubs, and so on, with the company’s approval and encouragement. Again, according to the company’s literature, PC-WRITE was found in at least one survey to be the sixth most frequently used MS-DOS word processor, out of a field of some 200 available packages.

PC-WRITE is Quicksoft’s only product; yet although it has always been marketed as shareware (in this country, that is; overseas, it is available only through regular retail channels with full documentation, registration and support and at somewhat of a premium over its American registration price), it has been quite profitable. The company’s gross has risen from a quite satisfactory $290,000 in its first full year, 1984, to $1.5 million in 1986.
The combination of Quicksoft, Bob Wallace and PC-WRITE may well be a phenomenon worth considering as a graduate case-study or thesis topic in marketing.

And finally, that status symbol of arrival in software’s jet set has appeared in chain bookstores: an independently written “guide” (priced about the same as the two-disk package itself) from a trade publisher; part of the literary cottage industry that supports—or is supported by—extraordinarily successful programs. In passing, however, it should be observed that neither PC-WRITE nor its clearly and engagingly written manual would appear to present the user interface problems of other packages such as WordStar and the dBASE family, around which large—if only occasionally helpful—bibliographies have grown.

As indicated, PC-WRITE may be had from a variety of sources at nominal cost, and the distribution disks may be copied and shared at will. Versions through 2.55, discontinued early last year, contained a complete manual of almost 150 pages on the single distribution disk. With Version 2.6, documentation on-disk was restricted to a brief introduction and reference sufficient to get the program up and running. The rewritten manual had to be purchased separately, or as part of registration. That is also the procedure with 2.7.

The reduction of on-disk documentation is probably a matter of simple logistics than of a change in business policy. Whereas Version 2.55 provided nine adequate but simply appointed help screens—fewer if one had only 128K of RAM—Version 2.6 included a vastly expanded file of 40 screens, accessed by cursor from a menu of topics in reverse video that filled the top half of the first “Help” display. Version 2.7 has added still more topics and screens, as well as spell check files. And the manual itself has grown to over 300 pages, typeset and illustrated.

Impressive. But how does PC-WRITE measure up as a scholar’s tool? Frankly, quite well for abstracts, relatively short articles and papers, and even book-length manuscripts, provided one keeps the chapters to within the 30-page maximum file capacity (or does not mind editing longer chapters in two or more chunks).

PC-WRITE’s editor has all the necessary features: macros, foreign accents, mathematical symbols and custom characters. Embedded ALT-key font commands, boldface, underscore, superscript, subscript, italic, footnote, index and table of contents capability, micro-justification, proportional spacing, and a number of other printer enhancements are supported. Defining a printer driver is as easy as choosing from a list of

Fig. 1: Sample “Help” Screen
nearly 400 presented by a utility during the initial "workdisk" setup routine. Cursor movement, block moves and deletes, search and replace are accomplished with blazing speed: at least as fast as that of the reputed champion, Xywrite. Single stroke "undo," variable margins, "bookmarks," mouse support, split screen, easy default modifications and customization are all there at one's fingertips.

PC-WRITE's weaknesses, if nowhere near offsetting its strengths, are nevertheless pronounced. Ironically, they stem from the basic premises underlying PC-WRITE's design: it should run on virtually all IBM compatibles, even the most tenuously so; that it should take a modest amount of RAM—128K, until quite recently, although full help screen utilization required more RAM in 2.55—and that it support the large number of printers. The trade-off was that the "Edit" and "Print" modules had to be separate, and although RAM has become far cheaper and the average machine's memory larger, PC-WRITE has retained its original configuration.

The editor is so good that the inconsistency between it and the formatter/printer is arresting, even after one becomes accustomed to the latter and able to get around and out of it what he or she wants. The program takes care of the transition between editor and printer smoothly and elegantly (assuming two drives or a hard disk), saving the file to disk, loading the printer and then reloading the data with but two key strokes; and then reversing the process once the print run is finished. The whole operation can scarcely be characterized even as a minor inconvenience.

But the formatter utilizes a somewhat cumbersome and archaic system of dot-command lines for control. With planning, practice, and occasionally some creative sleight-of-hand, all sorts of layouts are possible; they are HOT, however, necessarily convenient.

The second flaw, already noted, is the limit on file size that may be edited. This, too, is a function of PC-WRITE's RAM-thrifty roots. Regardless of on-board memory, no larger a file than 60K—about thirty pages—can be edited or loaded for printing. Files can be queued so that larger documents may be printed out; however, there is no question that PC-WRITE's limited data capacity causes its overall potential, particularly the speed of its editor, to be underutilized.

The main feature in the 2.7 upgrade, as mentioned above, is the spelling checker, including a master dictionary with 50,000 words and a separate user file for custom lists. The routine can be set to check a file or to flag misspellings as they are typed. It will display,
on demand, a “guess” list of similarly spelled words. Early versions of this feature seem to have offered some odd options, simply because a certain pattern of letters was ascertained. The current version (termed 2.71) seems to have ironed that bug out.

Without a doubt, the spell-check routine is elegant and handy to have, and I freely admit a measure of personal preference: I’m not fond of spelling checkers and make far less use of them (and thesauri, for that matter) than I probably ought to. But not something else have been more welcome in a major upgrade? All versions of PC-WRITE output files in ASCII (there is no “print-to-disk” function present or required), that are easily—and as it turns out in the case of 2.7’s spell-check routine, more quickly—checked by such programs as IBM’s Word Proof. Thus, it is arguable that Bob Wallace’s programming genius might well have turned to creating new routines where PC-WRITE lags behind its top-of-the-line competition: a redesigned in-memory formatter/printer and a greatly increased file capacity.

Certainly, such changes would take a new version of PC-WRITE beyond the capacity of some machines heretofore supported. In that case, consideration might be given to rereleasing an earlier version as a less powerful companion: say 2.6 or 2.55. The latter is still circulating, to judge by the bulletin boards in the Greater Boston area; indeed, it seems to have become somewhat of a “cult classic” with a loyal and affectionate following who love it for its clean lines, functionally ascetic help screens and spartan on-disk manual (and withall, its ease of use).

A final observation about PC-WRITE 2.7 and its predecessors. For me, there is a certain, hard-to-quantify, esthetic pleasure in writing with it: there seems to be a place for everything, and everything seems to be in its place. Nothing ever gets in my way; but everything seems to be an extension of my working habits. The editor’s commands are clear, intuitive and easy to remember. There is even a kind of satisfaction in the challenge of manipulating the formatter’s dot commands (and they are most certainly NOT clear, intuitive and easy to remember) and getting around the small file capacity. I would gladly do without those challenges were future upgrades to do away with them, but in the meantime they are tolerable flaws in an otherwise elegant program.

NOTES

1See Steve King’s review of PC-WRITE 2.7 in InfoWorld, 22 December 1986, p. 44.


John Ogasapian is professor of music history and literature at the University of Lowell. He has published three books, over thirty articles and numerous reviews. He also edits a quarterly in his field of specialization. John may be reached at P.O. Box 194, Pepperell, MA 01463.
Bibliography Update
Bradford A. Morgan


---------. “The Mac and the PC Communicate: How You Can Have the Best of Both Worlds by Linking a PC and a Mac and Transferring Files between Them.” PC Magazine. 6:3 (February 10, 1987), pp. 120-121.


---------. “Stacking up to the Mac: Desktop Publishing on the PC Must Match, and We Hope, Surpass, the Standards Set by PageMaker on the Macintosh. This Inspection of Output Samples Lets You Be the Judge of PC-based Page Composition Software.” PC Magazine. 6:3 (February 10, 1987), pp. 159-173.


Schneiker, Henry and David Palmer. "'Fit To Print: Interleaf's Do-It-Yourself Electronic Publishing Package Brings a Print Shop to Users' Desk Tops.'" *Digital Review.* 4:1 (January 12, 1987), pp. 73-76.


"Setting Type with TeX." *PC Publishing.* (July 1986), pp. 19-20.


---

**Russian Scholar Seeks Correspondence**

Dear American Colleagues:

We’d really very much appreciate your sending us your articles and the other printed matter in the use of computers in education. Novosibirsk University has been trying to develop computer-based instruction for several years, so it would be of great interest to find out how CBI is being developed in American colleges and universities. Could you send over this request to other colleagues who deal with CBI? Hope to establish good cooperation with you. Yours most cordially,

Dr. Yuri Tambovtsev
University of Novosibirsk
P.O. Box 124
Novosibirsk-90, USSR 630090

---

**October ESCC Conference in New York**

The Third Eastern Small College Computing Conference will be held at Marist College in Poughkeepsie, New York, on October 16-17, 1987. Abstracts of proposals for papers and panels were accepted until March 16, 1987, on general computer topics, including text analysis, writing, word processing, modern languages, humanities, and linguistics. Contact Dr. Barbara R. Sadowski, Director of Academic Computing, Division of Computer Science and Mathematics, Marist College, Poughkeepsie, NY 12601, or call (914) 471-3240.

---

**Wyoming Conference in June**

“Literacy in Schools and Communities” will be the theme of the sixteenth annual Wyoming Conference on English, to be held in Laramie on June 22-26, 1987. Papers had been called on various topics, including computers, until March 15, 1987. Contact Tilly Warnock, English Department, Box 3353 University Station, Laramie, WY 82071.

8—RWPH, March '87
Evaluating Student Papers With a Word Processor:  
A Progress Report

William J. McCleary

One of the problems with grading essays in writing classes and elsewhere is the amount of time it takes to write notes to students about the quality of their papers and what they should do next time to improve. Grading papers and writing such comments consume the majority of a writing teacher's time, yet writing evaluations by hand is usually ineffective. Because time is scarce, comments tend to be terse and cryptic. Also, an instructor's poor handwriting may interfere with the messages to students. Consequently, ways have been sought to use computers to provide more and better feedback to students.

In April, 1986, I was loaned a microcomputer to experiment with such a method, which involved evaluating student papers with the assistance of a word processor. The method was developed by Bradford Morgan of the South Dakota School of Mines and Technology and reported in the November 1985 issue of Collegiate Microcomputer. In addition to reading this article, I attended a session at a recent conference in which Dr. Morgan and two other speakers explained and defended the method. The following paper explains Dr. Morgan's basic method and how I implemented it with Genesee Community College's equipment and software—and with my limited computer literacy. It concludes with an evaluation of the method based on my admittedly brief use of it in late April and early May of 1986.

DESCRIPTION OF THE METHOD

Briefly stated, the method involves building up a file of "canned," or "boilerplate," paragraphs commenting on various aspects that may occur in student papers. For example, the file might contain three paragraphs commenting on the organization of the paper—that the organization was good, that it was so-so, or that it was poor. Each paragraph would explain in some detail what good, so-so, or poor organization, respectively, would be like. In similar manner, there would be series of paragraphs commenting on style, mechanics, logic, and other evaluative aspects of student papers. Then, when a teacher is grading a paper, he or she would choose the paragraphs that give comments appropriate to the paper being graded and would type into the computer a series of commands to retrieve and print those paragraphs. This way, each student will receive a page or two of such paragraphs customized to his or her own paper. See the end of this paper for an example of the commands written for one student's paper and what the commentary page looks like when printed out for the student. (Morgan's system also involves a record keeping program, but I only used the word processing part.)

As a result of this method, students receive far more feedback on their papers than any instructor would have time to write by hand. And the instructor can give all this feedback in about the same time as it would have taken to grade the paper in more traditional ways. The time required to write and enter the paragraphs is enormous, but to a certain extent the paragraphs would cover what a teacher would write anyway on various papers. If the teacher writes them on the computer, they can be saved and used for later papers. Once a file of comments is built up, future uses of the same set of comments shouldn't require further investments of time.

In addition, it is still possible to give students comments pertaining to their own papers. These comments can also be typed into the word processor at appropriate places between the commands to print the canned paragraphs. It is even possible to write the canned paragraphs in such a manner that customized comments can be appended to the paragraphs. Finally, the method is not limited to giving evaluative comments. Explanations and mini-lessons can also be canned and printed out on command.

IMPLEMENTING THE METHOD AT GCC

I experimented with the method in English 102, during the unit on scientific/deductive writing. In this unit, students must produce a deductive argument about an ethical problem. Specifically, they must find a news report of someone who faced an ethical dilemma (such as Michael Deavers and accusations of influence peddling) and write a paper judging the ethics of the person as objectively as possible. Since I have taught this unit many times before, and since I prescrie an organizational pattern for this paper, I found it easy, though time-consuming, to write evaluative paragraphs on the various parts of the essay and other paragraphs commenting on the essay in general. I also wrote some explanations and mini-lessons on various mechanical problems such as comma splices. Most of these were written before I graded any papers, although a few were written during the grading and then saved for permanent use.

The equipment involved was an IBM PC/XT with two

RWPH, March '87—9
floppey disk drives loaned to me by Computer Services. For the word processing program I used Volkswriter Deluxe. Below are the steps I went through in using Volkswriter Deluxe to evaluate papers.

PROCEDURE FOLLOWED

DOCUMENT FORMAT. I set up a standard format, using one of the methods for “How to Specify a Document Format Automatically,” pp. 3-13 and 1-14 of the Volkswriter Manual. I called my format VWFORMAT.GR and then used the GR extension for all files associated with this experiment. (In my mind, GR stood for “GRowe.”)

FILENAMEs. I gave each canned paragraph a mnemonic filename including an abbreviation and a number (plus, of course, the GR extension). For example, filename INT1.GR is the name given to the first paragraph I wrote commenting on a possible INTroduction that a student might have written. THE2.GR makes a comment on one type of THESIS statement.

FORMAT OF COMMENT PAGE. The standard format for a page of commentary included a heading (including the type of paper, the date, and instructor’s name), a space to enter the student’s name and topic, a heading for commentary on specific parts of the student’s paper, and a heading for general comments. Lessons on mechanical errors were to be included under general comments.

ASSEMBLING A COMMENT PAGE. Volkswriter’s “print” command, an embedded command, was used to recall paragraphs into the page of commentary. For example,.printTHE2.GR would recall and print out a paragraph about the student’s thesis statement. (I could have used the “read” command instead and thus actually composed pages of commentary that could be viewed on the screen. However, I was running short of disk space and creating entire actual pages would have consumed too much space.)

ADDING NEW PARAGRAPHS OF COMMENTARY. Volkswriter’s “block” and “notepad” commands were used to save new paragraphs that I wrote during the process of grading papers. That is, first I marked the paragraph with the block commands—F5 at the beginning and F6 at the end. Then I saved the paragraph with the “notepad” command, CTRL/F2. When the notepad command is pressed, the computer asks which file the paragraph should be added to. I ignored that question and made up a new filename for each paragraph, one that fit in with the previously created names. For example, I had two paragraphs about comma splices, CS1 and CS2. While grading a paper, I wrote a new paragraph, saved it with the notepad command and called it CS3. These paragraphs appeared in the directory as regular files.

CUSTOMIZING COMMENTARY. In addition to paragraphs of standard commentary, I wrote some paragraphs that customized comments could be added to. For example, I had a paragraph saying that certain facts had been omitted from the description of the case. I left myself a space at the end of this paragraph to add an example of a fact missing from a particular student’s paper.

ENDING A COMMENTARY. The PAGE command was used to end each student’s page of commentary. That way, I could enter commentaries for half a dozen or so papers and print them all at once. (I didn’t have a printer and so had to take my disks to the computer and printer in the department’s main office.) Using the PAGE command may have been a bad idea, as I will discuss later.

RESULTS OF THE EXPERIMENT

As expected, writing the canned paragraphs required an enormous investment of time before any grading could start. Also, the papers required more than the usual time to grade because I kept encountering new problems that I had no prepared paragraphs to cover. Then I had to stop and write new paragraphs; and if I wanted to save and reuse a paragraph, I had to use extra care in the writing since a reusable paragraph has to be simultaneously general enough to apply to future papers and specific enough to tell the student something about his or her own paper. I can’t even begin to estimate how long the writing took, but it was enormous—not something I could do in an ordinary semester with a regular load of classes and students. (A regular load is five classes at my school.)

The final collection of canned paragraphs is also large. A list of them is included at the end of this paper. Each filename is given, along with a summary of the paragraph that the filename represents. As can be seen, there are a lot of files, enough so that the file directory on the computer screen consists of two pages. Most of these apply only to the particular writing assignment for the scientific/deductive unit, but a few, especially those about mechanics, could be copied to another disk and used with other assignments.

These are not all of the canned paragraphs one would really need for efficient grading of this assignment. However, I feared that adding any more would use up the space on the disk and leave no room for making up the pages of commentary on student papers. It also became difficult to remember all of the files I had
available. Stopping to review my list of files or to find the right filename used up the time I was supposed to be saving by grading with the computer’s help.

However, I should note that grading scientific discourse requires more commentary than any other kind of writing assignment. Grading papers written in other aims of discourse would probably be a lot easier, and would require fewer canned paragraphs.

There were other problems with the method as well. The biggest was that I have an IBM computer only here at school. My computer at home is not compatible, so I couldn’t use it to implement this method. Yet, like most English teachers, I do more paper grading at home than at school. Consequently, I ended up grading only about 15 papers with the computer. The rest I graded in the traditional way, at home.

Another problem was that even grading papers at school was difficult because of the desk space required by the computer. After putting this big computer on my desk, I hardly had room to lay papers down to read and mark them, to say nothing of what I was supposed to do with my other paperwork. I ended up with little stacks of mail, memos, monographs, and other papers all over my office—on the floor, the filing cabinets, the typewriter, etc. I needed a bigger desk or a smaller computer.

Finally, I kept encountering problems that required much time to solve, plus some that I never did solve. For example, the status line at the bottom of the screen stopped giving me page numbers and only gave total accumulated line numbers instead. I suppose this occurs because of the PRINT or the PAGE command, but I haven’t been able to find out for sure. Also, the printer would quit at the last line of printing and would not roll the paper forward to the end of the sheet. I don’t know why. Nor do I know why there was no 6-line margin at the top as specified by the format. I determined the answers to a few questions like these, but it was just taking too much time to track down every detail. I used informal solutions instead.

Lest I make the picture sound too bleak, however, I should note several unanticipated but fortunate discoveries. First, writing these paragraphs forced me to think through my criteria very carefully, something that every teacher should do more of. Second, the criteria were then applied much more consistently to student after student. After all, unless I was to keep writing new paragraphs, I had to confine myself to what was already written. Third, though I couldn’t talk any other faculty members into trying the idea (since most either cannot type or cannot use a computer), they were fascinated with my list of canned paragraphs.

Everyone teaching the same course asked for a list of them to use as a reminder of what they were supposed to be looking for while grading the papers. We probably achieved more consistency in grading by different instructors than ever before. Fourth, I became hooked on the method. This fall when working on rough drafts of term papers, I found myself writing the same comments repeatedly and thought longingly of computer-generated comment sheets. And I saved copies of all my comments for a time when I can experiment with computer grading again. Finally, I discovered that the method works just as well, and perhaps even better, to help students with preliminary drafts as with final drafts. This would make sense in view of what has been learned about the “process approach” to teaching writing.

CONCLUSIONS

My main conclusion, after investing all this time, is that Bradford Morgan’s method does show promise. If an experienced faculty member spends the time to write and enter all or most of the canned paragraphs, not only that faculty member but all others who teach the same course could make use of the paragraphs. Secondly, because the paragraphs were created by word processing, every instructor could revise them to suit his or her own tastes. And, as promised, these paragraphs could provide students with more commentary about their papers than can be given in any other way. The commentary would be more legible, too.

In the future, though, the commentary ought to be provided more often on students’ rough drafts than on their final drafts; I think both Morgan and I made a mistake in this respect. The commentary should be more usable when students can immediately implement the evaluations as they rewrite their papers. Of course, comments for rough drafts would need to be worded in a slightly different manner than those for final drafts.

However, I discovered that the “hardware problems” may be the major stumbling blocks. For instance, the method will not reach its full potential until every instructor has a computer in his or her office, set up in such a manner as to leave plenty of desk space. Having one computer in the central office (as we do presently, in common with nearly every English department I have visited) will not do the job. Instructors will also need to be networked into a printer so they do not have to constantly remove their disks from their own computer and take them to a computer with a printer. Also, the computers may need hard disks to alleviate the problem of running out of disk space.

As for the problem of incompatibility with one’s home computer, we may never get around that. But writing teachers spend far too much time grading papers at home anyway. Here is a case where technology might
force a beneficial change of behavior on its users by forcing them to do their work while at school. But first instructors need computers.

Realistically, the only way a college administration is going to provide a computer for each writing instructor is if the computer allows each instructor to teach more students—i.e., to become more "productive," as administrators measure productivity. (Yes, I know that faculty members define productivity in terms of how much they teach each student, but teaching each student more content brings in no more money to the institution. You need money to buy computers, and that comes from teaching more students, not more content.) The jury is still out on whether grading with a computer can increase either kind of productivity.

William McCleary teaches in the Department of English at Genesee Community College in Batavia, New York 14020.

---

**Manuscript Submissions Welcome**

The *Newsletter* welcomes article submissions that pertain to word-processing, text-analysis, and research applications in professional writing situations. Also, hardware and software reviews are accepted, but please contact Dr. Jim Schwartz, Hardware/Software Review Editor, before submitting them (call Jim at 605-394-1246). Manuscripts either may be submitted as hard copy or on 5½" diskettes using WordStar, MultiMate, DCA, or standard ASCII code. If submitting disks, please make sure they are formatted either in MS-DOS, PC-DOS, or a popular CP/M format (Kaypro, Zenith, etc.) The Editors reserve the right to edit manuscripts, if necessary. If you want your manuscript or diskette returned, please send enough postage to cover the return along with a self-addressed envelope. Address all correspondence to the Editors, *Research in Word Processing Newsletter*, South Dakota School of Mines and Technology, 501 E. St. Joseph, Rapid City, SD 57701-3995. The Editors may also be reached on CompuServe (70177,1154).
GENESEE COMMUNITY COLLEGE

ENG 102 Writing Assignment for Scientific Deductive Discourse:
Writing about an Ethical Issue

Due Date: April 25, 1986. Instructor: Bill McCleary

Student: John Doe Topic: Elizabeth Bouvia's desire to die

Overall, this is a pretty good essay. The topic is good, the organization follows the assignment, and the argument is mostly sound. Below you will find some comments on specific parts of your essay, followed by some general remarks on the essay as a whole.

COMMENTS ON SPECIFIC PARTS:

Your introduction does bring up the general issue, as it is supposed to, but the issue is not very clear. Perhaps you could have made it clearer by mentioning a well-known example or two of someone who has been involved in this type of issue.

Your review of the facts is fairly complete, but there are a couple of missing facts, and the lack of these facts makes the argument incomplete. The missing facts are: That the hospital thought she wanted to kill herself by starvation.

COMMENTS ON THE ESSAY AS A WHOLE:

You have one or more comma splices of a particular kind. As you probably know, two independent clauses should not be joined just by a comma. You need a comma and a conjunction, or a semi-colon. Words such as "therefore" and "however" are conjunctive adverbs and do not count as conjunctions; therefore, you need to put a semi-colon before such a word when the word appears between two independent clauses. Here are a couple of examples:

John is right this time; however, he is usually wrong.
A butterfly is fragile; therefore, it must be handled with care.
The rain fell for two days; consequently, the river overflowed.

You have one or more errors of pronoun agreement. Whether a pronoun should be singular or plural depends on the noun being referred to. (Remember, pronouns "stand for" nouns.) The noun being referred to is called the "antecedent," and the rule is that pronouns are supposed to agree with their antecedents. The most common error is for the noun to be a singular term like "a person" or "an individual" and then for the pronoun to be a plural form like "their" or "themselves." An example would be as follows:

WRONG: A person should always take their raincoat to school.
RIGHT: A person should always take his or her raincoat to school.
or People should always take their raincoats to school.

NOTE: When the sex of the person is unknown, the correct singular form used to be the masculine: "A person should take his raincoat to school." However, the construction has been criticized as being sexist, so most publications prefer one of the two versions listed above.

AWPN, March '87—13
LIST OF GRADING COMMANDS

SCIENTIFIC/DEDUCTIVE ESSAY ON AN ETHICAL PROBLEM

INTRODUCTION
INT1 - general issue too general
INT2 - general issue too specific
INT3 - general issue hard to understand. Example needed.

REVIEW OF THE FACTS
FACT1 - some facts used in body not in this review. An example is:
FACT2 - much too long. Leave out stuff not relevant
FACT3 - missing facts. An example is:

SPECIFIC ISSUE
SPEC1 - specific issue missing
SPEC2 - not clear enough; should mention who, the act, the field.

THESIS
THE1 - thesis is missing
THE2 - thesis doesn't take a clear stand.

DIRECT ARGUMENT
DA1 - missing rule. Rule should have been:
DA2 - missing evidence statement. Should have been:
DA3 - missing definition. Term needing definition is:
DA4 - missing backing for rule. Should have been:

FIRST COUNTERARGUMENT
1CA1 - too short. Give opposition more credit. Example of more:
1CA2 - right idea but said poorly. Should have said:

FIRST REBUTTAL
1REB1 - repeated the direct argument. Argument could have been:
1REB2 - oversimplified. Should have dealt with this issue:
1REB3 - missing rule. Rule should have been:

SECOND COUNTERARGUMENT
2CA1 - Too short. Give opposition more credit. Additional part:

SECOND REBUTTAL
2REB1 - Repeats direct argument. Issue addressed should be:
2REB2 - Too short to defeat the CA. Another part could have been:
GENERAL COMMENTS
GEN1 - Title, type of paper, subject of paper
GEN2 - intro comment; pretty good essay. Comments below. (incl. subhead for specific comments)
GEN3 - COMMENTS ON THE ESSAY AS A WHOLE
GEN4 - intro comment; essay not too bad but has some major problems
GEN5 - lack of coherence, esp. transitions
GEN6 - good coherence
GEN7 - completeness problem; good DA, hardly any CA or REB
GEN8 - completeness problem; a lot of missing rules of interpretation
GEN9 - completeness problem; oversimplified CA's and REB's.
GEN10 - completeness problem; one strong CA missing. It is:
GEN11 - bad topic; either little controversy or not really an ethical problem
GEN12 - connotative language
GEN13 - CA's better than REB's
GEN14 - Type 3 problem, when Type 2 was assigned
GEN15 - Didn't hand in all the requirements
GEN16 - Intro comment. Poor paper. Follows outline but that's all.
GEN17 - too many errors; grade has been lowered because of them
GEN18 - Watch that spelling; most noticeable error; makes you look bad

CS1 - comma splice; first notice, with examples
CS2 - comma splice; second note, see teacher
CS3 - comma splice; conjunctive adverbs (e.g., however, therefore)

AGR1 - errors in pronoun/antecedent agreement (number)

COMMA1 - errors in commas for nonrestrictive clauses (explanation)

FR1 - some fragments OK; most of yours are not
FR2 - fragments not acceptable in referential writing

SC1 - too many short sentences; sounds choppy; combine into longer ones
In Future Issues...  

WordStar 4.0

MultiMate Advantage

More on Desktop Publishing

Hercules Graphics Plus Card