DOCUMENT RESUME

ED 308 878 IR 052 875

AUTHOR	Cinque, Harriet A.; Santomauro, Dennis J.
TITLE	Information Literacy: Application Software and Critical Thinking. An Experience and a Proposal.
PUB DATE	7 Oct 87
NOTE	18p.; Paper presented at the National Educational Computing Conference (Dallas, TX, June 15-17, 1988).
PUB TYPE	Viewpoints (120) -- Reports - Descriptive (141) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	Business Administration Education; *Computer Literacy; *Critical Thinking; *Curriculum Development; *General Education; Higher Education; Microcomputers

ABSTRACT

This paper describes the development of a course to introduce microcomputer application software to non-computer science majors at Kean College (New Jersey). Topics covered include: (1) background; (2) course content; (3) classroom laboratory design, installation, and management; (4) teaching strategies and student performance and reactions. Some related conceptual issues are then discussed, and a proposal for offering computer literacy as a general education course is presented. The paper concludes with discussions of computer literacy, the rationale for incorporating critical thinking skills in the curriculum, the relationship between application software and critical thinking, and problems with considering the course as a general education offering. (21 references) (MES)

INFORMATION LITERACY :

APPLICATION SOFTWARE AND CRITICAL THINKING

AN EXPERIENCE AND A PROPOSAL

Submitted by: Harriet A. Cinque
Department of Mathematics and Computer Science
Kean College of New Jersey

Dennis J. Santomauro
Department of Mathematics and Computer Science
Kean College of New Jersey

Date: October 7, 1987

Abstract

Since microcomputers have become easier to use, we believe that the early introduction of application software packages to non-computer science majors effectively develops student productivity in his major. Accordingly, at Kean College we offered two pilot sections of such a course to management science and accounting majors after we had been awarded a New Jersey Department of Higher Education Grant in Computers in Curricula Program. This paper is divided into two parts; the first of which describes our experiences in developing and presenting the course. The second is more conceptual in nature and suggests how critical thinking skills could be developed through a restructured computer literacy course utilizing application software and suggests a future direction for general education.

PART I: THE COURSE

Background

At the college, management science and accounting majors are required to take a two-course sequence in computer science. These courses were designed to provide the business major with an introduction to the computer and an understanding of the nature of programming and the problems encountered by the programmer. The first course was designed as a broad-based computer literacy course while the second had as its major objective developing in the student both an awareness and an understanding of data and file manipulation in a business environment. This approach was appropriate for an era when computer systems within business and industry were largely centralized and individuals in areas of functional management relied on members of computer staff to provide the information necessary for planning and decision making.

However, the introduction of the microcomputer and the widespread availability of application software have changed the work place. Computer systems are smaller and have become decentralized and directly accessible to people in all functional areas of business. With this in mind, we examined the two courses.

The first course seemed appropriate whereas the content of the second did not. The former had been revised in the preceding year and was being taught in a laboratory equipped with IBM PCs. The latter focused upon the need for the business person to communicate his or her data requirements in programming terms. The language RPG was utilized and accessed on one of the college's centralized computing systems, a PRIME 9750. This second course was restructured to reflect the changes in the business world with respect to the microcomputer.

The specific objectives of the revised course center upon creating a student awareness of the importance of the microcomputer as a management tool as well as equipping the student with the knowledge to adapt readily to changes in technology in the work place. Tantamount to achieving these objectives is the need for as much student "hands-on" as possible.

Since the management science and accounting majors comprise approximately 2500 students, the largest major at the college, our primary concern was the lack of hardware. The route chosen was to apply for funding under the State of New Jersey Computers in Curricula Grant Program. We submitted a proposal to equip two microcomputer laboratories, one as a dedicated classroom and the other for student use outside the classroom. Each lab would include 15 IBM PCs and a printer and the necessary software and documentation. Also included was the request for LAN support as well as personnel. The grant was awarded and funded at $75,000.(We had requested $87,000.) Subsequently, we revised the curriculum and offered two pilot sections in the Spring semester 1987.

Course Content

We developed the course to address the role of the microcomputer "in the development and organization of data in files and databases for information generation." It accomplishes this through the study of "the selection and proper use of microcomputer application packages to fulfill the information needs of business and support management problem solutions" [18]. It acknowledges that with the rise of the end-user approach to solving problems, the emphasis has shifted from the tools used to analyze to the problems being analyzed. Hence, there is greater emphasis on the nature of the problems themselves. Since this is the second course in a year sequence, it has more depth than that of an introduction or survey. Topics include:

-a review of computer system components
-an in-depth study of the nature of data including its structure, organization, storage and operations
-concepts and functions of information systems
-categories of software
-systems and program development processes
-end-user application development
-software packages
-communications
-system integrity
-social and ethical issues.

Since we felt that the use of the 'correct' text would contribute greatly to the success of the course, we conducted an extensive search during the semester prior to the pilot. We wanted a concept approach; hence a 'generic' text. We knew also that some form of lab manual was a necessity. The texts we examined with the exception of the one finally chosen seemed to fall into three general categories for this type of course:

(1) the survey introduction to computers/information processing compendiums of a little of everything without depth in anything
(2) the cookbooks of package skills, heavy on the 'how-to' but light on the 'why?'
(3) the templates for solution of specific applications.

Since the introduction to computers fell in the realm of the college's first service course, we shied away texts in category (1); the remaining two categories were judged too limited in scope.

Late in the term, a new text was published which spanned the survey and templates categories by offering a comprehensive view of the tasks microcomputers can handle through case studies; these were embedded in a generic solution model focusing on overall concepts rather than specific packages. We selected this text: Microcomputer Applications by R. Grauer and P. Sugrue. Its title is perhaps misleading since it approaches the subject from the data management point of view. We are satisfied with our choice. An exit survey after the pilot sessions confirmed our view; students found it interesting, appreciated its contemporary flavor and enjoyed reading it. An aspect of the book appropriate for management science students is the review of the computer science industry through a series of corporate profiles on various companies and individuals in the field.

Classroom Laboratory

Design

A conscious decision was made to integrate the classroom and laboratory as opposed to the usual operation of computer science courses which separate the two. The microcomputers were placed "galley style" along both sides of a classroom; the instructor at one end of the room can easily monitor all the screen displays with a sweeping glance. This physical layout was an advantage while conducting "package feature walkthroughs"; one could easily tell a student's progress by observing the screen in relation to all the others.

Advances and interest at the college in the area of networking prompted the decision to install the microcomputer equipment under a local area network. A token ring configuration was chosen; its requirement demanded groups of seven machines per Multiple Station Access Unit and as a result a network of 14 machines was created in the classroom. Since the same classroom was to function as an open laboratory during non-class times, it was decided to place the server and the printer in an adjoining room for security reasons. This setup limits the instructor's access to the server during class time; however, this caused no difficulty since the class was not studying networking per se.

The software packages selected for study were the full application packages used in industry not limited student versions. Having the network eliminated the problem of handling the multiple sets of floppy disks required for operation of several application packages. This factor alone justifies network use. The lab assistants found the task of controlling the distribution of only 14 diskettes (those containing network files) time consuming and frustrating. It is fair to say that diskette handling would probably be the biggest problem in the operation of a microcomputer laboratory.

Usual class size in computer science courses is 30 students. Obviously, with 14 machines available, doubling up (and even, tripling) was necessary. The ratio of two students to a machine was comfortable and even proved productive because of student interaction. We anticipated the benefit of peer interaction in advance since this has been proven a distinct learning technique [19,20]. However, more than that produced a crowded situation. In the future, we plan to limit class size to 28 and maintain the 2 student : 1 machine ratio.

Lab Installation and Management

The operation of the laboratory was functionally separated from the presentation of the course, thereby 'freeing' the involved faculty from the time consuming responsibilities of lab and network management. In order to do this, knowledgeable personnel are needed in the form of a network manager and a staff of assistants. After our experience with the pilot for a semester, we are convinced that any network undertaking in the collegiate, or for that matter any, environment requires a staff dedicated only to 'servicing' the network and the physical environment housing that network.

A graduate from our own computer science program became our network 'resource person'. He had participated in a department project studying network installation and was familiar with network operations and problems. Since his experience exceeded that of the faculty involved with the course, his assistance with the design and the development of the network was invaluable. His position was made available by state funds solicited for a Program Improvement Proposal. In effect, he functioned as our network manager and had sufficient background and expertise to revise the network program. He provided a network menu which listed all the software and packages available for use in the laboratory.

Lab usage totaled 15 hours weekly: 2 hours daily open lab (10 hours) and two sections meeting twice a week for 1 1/4 hours (5 hours) in the lab as a classroom laboratory. The open lab and laboratory classroom necessitates the presence of lab assistants whose functions are to:
(a) set up, maintain and service the equipment
(b) assist students with the use of a package (but not to do their homework assignments!)
(c) monitor lab usage
(d) control distribution of diskettes interfacing with the network
(e) assist the instructor during the lab portion of the class.

Towards the end of the term prior to the pilot course, we started to set up the team of 4 student lab assistants. The ideal lab assistant would be a junior with some microcomputer background in hardware and application packages. They were difficult to find especially when other jobs are available. We finally got off the ground with some computer science students with a PC background and others with good academic credentials but very little PC background. Few had any extensive experience with the packages but all demonstrated the willingness and capacity to learn.

Our 'resource person' prepared a training program involving hardware and software. He utilized the students to set up the equipment in the lab. On the whole the lab assistant staff was dedicated and dependable and functioned very well under the auspices and training of our 'resource person'. They demonstrated an allegiance to their position and the lab. That allegiance is a necessity when students are working with students. The lesson learned was that solid tight management under a single person responsible for the entire program is required for a well-functioning student laboratory. This supports the findings of Spence and Windsor [21].

In the fall term following the pilot we have 14 sections taught by 8 instructors. A second microcomputer laboratory with equipment made available through grant funds was installed during the summer. It is now our classroom laboratory. Lab usage is obviously heavier. The original lab is now the open lab. It functions 14 hours a day and 5 hours on Saturday for a total of 75 hours a week. We again encountered the problem of a limited pool of students available as lab assistants. To remedy this, we added the students who had completed the pilot course to the pool.

Teaching Strategies

When the classroom is also a laboratory, an approach beyond the lecture technique is a necessity. Not to use the equipment during a class would be wasteful of the resources. Initial ventures alternated between a short lecture followed by a short hands-on session and then an attempt to return to the lecture format within the same class session. We soon found that once the student's attention was directed to the hands-on exercise, it was almost impossible to continue with lecture since the student's interest was diverted to the problem on the microcomputer. Accordingly, class sessions are structured to lecture first followed by a directed laboratory session under the instructor's guidance.

Since there are only 14 machines and twice that amount of students, 'teams' of students work at each machine. Each student is required to provide a work disk and bring it to each class meeting. There is a tendency for the same student in a team to be the 'keyboarder' during the class. The instructor should oversee the alternation of students at this function.

The presence of the instructor in the lab environment necessitates a varied approach from that of merely following a lab manual. It offers the opportunity of the 'guided walkthrough' where the instructor can demonstrate various package features and usage techniques to all students at once. Hence, the lab manual for the course need not be as complete or detailed as one used in a course where lab is an independent venture for the student. When working with application packages during the pilot, we employed several approaches to the lab portion of the class. This was an attempt not only to determine the most effective one but also to determine the appropriate format for the accompanying lab manual. In lab sessions, we varied the amount of explanation of the features and the extent of the laboratory directions. Students favored some explanation and demonstration prior to their efforts in the lab and preferred lab directions without verbose explanations. Since we were using an application package for word processing that was not covered by existing lab manuals and we were not satisfied with the other portions of lab manuals, we began to develop our own set of lab procedures based on these findings.

Lab manuals provided with textbook packages are another complete area of study but several points are worthy of mention here. When reviewing a lab manual for possible adoption, remember that it will be utilized in two different ways. First, it will be a guide through the initial experience with the feature being studied. Then it will also be used as a reference for subsequent use in the lab. Too often, lab manuals are extremely good at providing direction for the first use but fail miserably as a reference. The same directions which were very detailed and complete for introductory use covering each step of the way become extremely frustrating for students searching for a specific direction. Identification in the form of headings with referral 'tags', paragraph names or in-margin referral points are a necessity. A lack of defined referral points cannot be compensated for by merely including a list of commands or directions in the back of the manual.

Another point to be considered is whether the particular package should be taught functionally (how to use it) or conceptually (the form of the package and what the package can do). With management science and accounting majors, the functional approach appears appropriate in the beginning. After some experience with package features in general, a combined approach may prove acceptable. Time constraints did not permit further investigation during the pilot.

It should be noted that as the students finished one package and proceeded to study the next, the time involved with introduction decreased. The instructor could make use of the features of the earlier packages studied as reference points for comparison. The more packages studied, the more they became familiar with the features employed by all packages to achieve a successful user interface. They had acquired sufficient knowledge and experience to become critical of package features toward the end of the course. Indeed, this was one of the goals of the course.

As with any computer science course, the students were assessed on a combination of tests and laboratory assignments to determine their grade. Various modes of testing were employed from traditional to hands-on. The subject of testing should be reevaluated as well as the mode of testing for this course. The traditional test (pencil to paper) is definitely appropriate for evaluating theoretical type questions. However, since a fair amount of learning in computer science is experiential, so also testing should be partially experiential. Many times a student cannot immediately provide an exact answer to a question addressing the specific use of a package on a traditional test. But having the microcomputer and package available for use allows him the opportunity to 'evoke' the answer. For example, in the pilot section on word processing, a portion of a test was conducted on the microcomputer and consisted of "hands-on" revision of a document provided by the instructor. It was not a lengthy exercise but it truly measured the student's ability and mastery of the tasks involved. The results of a student's efforts were obtained by printing/viewing the document on his work disk. This is a lengthy and time consuming task for the instructor. More research must be done on automating testing systems which can appropriately measure deviation from the desired results of package use.

We evaluated the pilot and, as a result, formulated several future enhancements. The demonstration of an on-line information search would give firsthand experience with communications. The study of business graphics and information presentation techniques will be extended. The use of videotapes and on-line tutorials will be investigated. Another evaluation for improvements will follow after the first offering of the course to all students.

Observations

Throughout the pilot offering, we were cognizant of student activity during class as well as lab. It was a way of evaluating student reaction to the course and informative in indicating areas of concern. Some of our observations follow. It is generally what we expected, although there was some variance at times.

-Some students assimilated the material very quickly and moved on at their own faster pace. Prior exposure to PC usage was not necessarily a factor.

-Initially there was very little response to information requests made by the instructor. This was somewhat surprising since some of the material was review from a prerequisite course. We found a mixed background since some students had taken the prerequisite in high school or at another college some time ago. This may also have been due to the fact that students were not expecting this revised course.

-There was difficulty with association skills. An example was not associated with a concept until they actually did it.

-Analysis and synthesis of information had to be directed when working with information stored in a database or playing 'what if' with a spreadsheet. It was obvious that students did not have much experience in this area.

-Creativity raised doubt and required reassurance. Students stumbled over minor things such as filename selection unsure of what was required and had to be encouraged to 'play' with the packages to get a feel for what they were capable of doing. Once encouraged to act freely, they were more relaxed about working with the packages.

-Student satisfaction with the course was generally high and there was an obvious level of interest which is frequently lacking in required courses. The students felt they were learning something useful. Soon into the first major topic, word processing, students began to

do work for other courses and during the spring break, several students came in to work on term papers.
-There was a significant amount of peer interaction. Students willingly helped each other and did so before requesting help from the instructor or from a lab assistant.
-The atmosphere in the open lab was positive overall. An esprit de corps developed in groups of students and progressed to the entire room.
-Students experienced a sense of accomplishment when they were able to design and execute a solution to a problem. We were surprised that their sense of self-satisfaction, at times, was independent of the correctness of their solution. Using the software to solve a problem seemed as important as, if not more important than, the results.
-Confidence in use built as usage increased but not at the same level for all students.
-It was obvious not only to the instructor but also to fellow students which students had neglected to 'practice' in the lab.
-Student frustration was evident at different times. The student would know what he wanted to do and was searching through a copious amount of text to find it. A higher level of frustration occurred when a student did not know "where to start" to solve a problem.
-Students were recommending the course to others and commenting "everybody should take this course". A few students who had already received a grade for the pre-revised course requested permission to register for the revised version as a free elective.

PART II: THE PROPOSAL

The Computer Literacy Question

There have been numerous articles written concerning the concept of computer literacy. Everyone realizes its importance and believes it should be an integral component of the educational process. However, it is a concept that seems to elude defining (this should not seem surprising since computer science, in general, eludes definition [2]) and, as such finds very diverse and varied implementations. Some universities, such as Carnegie-Mellon, have implemented a computing skills requirement for all students [9].

One definition for computer literacy that appears to be on the mark is from Anderson [1]: "Computer literacy is best defined as whatever computer knowledge and skills one needs to function effectively in a given role. This includes the ability to evaluate appropriate uses of the computer, to plan and execute various applications of the computer, and most importantly, the ability to understand how computers are impacting us socially, psychologically, culturally, and ethically."

It is apparent in the literature [3,4,5] as well as the plethora of textbooks focusing upon microcomputers that the computer knowledge and skills one needs to function effectively center upon application software rather than programming. Many colleges and universities have implemented such a course and we at Kean have done the same. Although our course has a dedicated population, we believe that the content material could be restructured and presented in a manner that would develop critical thinking skills for all students, irrespective of their major. We propose such a course be required for students in all curricula and that it should fall under the rubric of general education. To support our proposal we examined some of the literature in both education and computer science.

Rationale

A strong component of all collegiate education is the cadre of general education courses required for all majors. All disciplines build upon this componant and ours is no different [6]. Two of the most important aspects of general education that should be threaded throughout the coursework are problem solving and critical thinking. They should permeate all baccalaureate curricula. Professor Arnold Arons in his paper, "Critical Thinking and the Baccalaureate Curriculum", states a few of the thinking and reasoning processes that underlie analysis and inquiry involve consciously raising the questions, "What do we know ?", "How do we know ?", "Why do we accept or believe ...?", "What is the evidence for ...?". He further explains that the "How do we know ?" should be an intrinsic part of general education.

Isaksen and Parnes [10] emphasize that "effective problem solving is based upon both creative as well as critical types of thinking". They point out that creative thinking involves making and communicating meaningful connections, thinking of many possibilities and alternatives, becoming aware of problems, deficiencies, and gaps in knowledge, bringing together available information, making hypotheses and modifying and retesting them, and finally communicating results.

There is little disagreement among college faculty that students' thinking abilities must be improved; however, the discussion continues concerning the most effective means to attain this end. Some colleges teach critical thinking as a separate course while others incorporate it into general education and thus across all curricula; the latter being the method used at Kean. "Regardless of the route taken, critical-thinking experts say there is no foolproof way to teach thinking. What is certain, they say, is that faculty members must be ready to change the way they approach the learning process. Most thinking activities shun traditional teaching" [8]. However, writing assignments, in particular "freewriting", are very significant for teaching thinking since the student must clarify his/her thoughts before writing them.

Professor Robert Sternberg of Yale University explains that students need help in recognizing problems, not just in solving them [11]. He describes possible ways of teaching critical thinking and states "good thinking in one academic or particular area of endeavor does not guarantee good thinking in another". This is supported by a recent article examining the relationship between programming and thinking [13]. Professor Sternberg continues "critical thinking programs need to sample a variety of thinking skills and to sample them in ways that are true to the way problems appear in our everyday lives" [12].

The Application-Software-Critical-Thinking Connection

If we consider an application software course in which the computer is presented as a tool, it is clear that students will develop the lower level thinking skills of knowledge and understanding concerning how to use the specific software; namely, entering data, searching and sorting files, inserting and deleting text, etc. However, critical thinking as we have seen requires higher level thinking skills. We believe these can be developed by presenting application software, such as word processing, spreadsheets, and data base, in such a manner as to examine both textual and numeric information , formulate an analysis, and answer appropriate questions. Let us consider a few examples.

Using word processing facilitates freewriting and thus enables the student to develop his ideas with greater fluidity due to all the editing features. These features ease the rewrite process which, in turn, allows the student to fine tune his thoughts. Just as one becomes a better reader by reading(Duquesne University uses a specialized software package in teaching critical reading [14]), one also becomes more proficient at writing (and thinking) by writing. The facilities of word processing for reorganizing and synthesizing paragraphs to develop a topic should assist the student in fostering better writing skills. An instructor could easily present an unstructured essay and have the students use the word processing features of insert, delete, move, copy, etc. to create a well-developed expository one. Comparisons of text can be readily viewed. Outlines can be dynamically expanded within a document. The software becomes a pedagogical tool.

In addition, if the student has had some exposure to programming (many freshmen enter college with such an experience), the parallel between writing and programming should be explicitly noted for the student. They are similar inasmuch as both are creative, mental activities involving definition (topic in the former and problem statement in the latter), analysis, structure and syntax, stepwise refinement, and documentation.

If we wish to perform numeric analysis and formulation, then electronic spreadsheets provide a quick and easy means. In particular, consider a budget analysis using a spreadsheet. Every student usually has a working knowledge of a budget and realizes the implications inherent in one . Also, budget-type problems lend themselves to spreadsheet analysis. "Additional advantages to using spreadsheets include 'what if' experimentation. Students can alter values searching for the "best" solution to a problem." [5] They are able to examine the various parameters in a problem and to explore the connections between them as well as test a hypothesis concerning the relationship. The immediate feedback factor supports student exploration. "Try it and see what happens' becomes a natural expectation and initiates an analysis. Consequently, the students become involved with creative thinking.

This type of numeric analysis does not require a high degree of mathematical sophistication; high school algebra is sufficient. In addition, the student who has a history of "math anxiety" is free to explore.

When we have students working with data bases, they begin to examine the relationship between data and information. If we design the appropriate questions, the student will formulate a response that was generated by data base analysis. Hypothesis-testing is possible and discussion of what is relevant data should occur.

Consider a data base of the fifty states. The level of thinking skills vary depending upon the type of questions posed. To list all the state capitals is very low level and generating the ten most populated states is slightly above that. However, if we wish to know which ten states have had the greatest population shift in the last ten years, the level increases considerably. It does so even more when we look to the data base for some possible reasons for these population shifts. The student examines the data with a more discriminating eye. Questions arise as to what type of information can you get from a data base. Consideration is given to the design of the given data base and what would one need to obtain from the data base to test a hypothesis. All of this involves the higher-level thinking skills of analysis and evaluation. Dr. Janet Parker states very succinctly in her paper, "Tools for Thought" [illegible], "Getting data from a data base or spreadsheet is only the beginning; interpretations and inferences need to be drawn. Students need to be pushed beyond simple knowledge of the data, toward analysis, synthesis and evaluation. You will have to work to take students beyond the level of simple data input, recall and listing, toward evaluating what they see, making inferences about what it means, and coming up with some kind of meaning in terms of solving questions or problems."

The General Education Problem

We see several problems arising with a course that covers the content that we have described being considered as a general education offering.

First and possibly foremost, is this really a general education course that academicians can feel comfortable offering for the broad population of students? We believe it is and it manifests a twofold benefit. The faculty member has a new tool with which to approach the learning process and technology (although low level) is introduced to all students. The student gains elementary skills with this introduction, but more importantly begins to consider and analyze the information generated within a technological society.

Another difficulty lies in determining which faculty will teach this. Let us immediately state that it should not be the computer science faculty. This course is neither a service course nor a software survey course, but rather should be viewed as a component of general education. We support the statement that the use of applications packages should be considered a collegiate rather than departmental responsibility.[16] At Kean College the faculty have developed the following core of six GE courses:

- Composition,
- Inquiry and Research,
- Emergence of the Modern World,
- Intellectual and Cultural Traditions of Western Civilization: 1450 to the Present,
- Landmarks of World Literature, and
- Science and Technology in the Modern World.

These courses are taught by professors from varied disciplines and from different schools within the college. It is our belief that the proposed course should be handled in the same manner. However, concern may arise about the qualifications of an individual faculty member to teach this. We believe most faculty will not be prepared to teach this outright and that some training and workshops would be necessary, especially to insure that the course does not deteriorate to acquiring only low level skills.

We agree that computers and their software are not neutral tools and that they have the capacity to amplify as well as reduce different aspects of learning.[17] That is why this course cannot stand alone. It must be prefixed by or collateral with other courses that develop problem solving ability and analytical reasoning. (At Kean the thrust of the new GE core is just that.) Some may suggest that the content we describe could be incorporated within another course. We caution against that since we feel that the material will not be given adequate coverage from both the critical thinking and skill-based perspective.

Since the course would require "hands-on", students need ready access to equipment. This may not be a problem at some institutions, but would put a definite strain on those with limited resources. The entire issue of labs raises many concerns involving staffing, maintenance, availability of software, etc. These are problems that must be addressed college wide and require administrative decisions.

Summary

Our experience in teaching information processing to management science and accounting majors via software productivity tools has led us to the belief that more can be accomplished. Processing information and using software to assist in problem solving is relevant to all students, not just a select few being serviced in a specific major. Thus we have set forth a proposal for a general education course that may raise many questions for which we have no ready answers. However, we believe it is worth considering. No attempt has been made to present an outline for such a course since the proposal itself warrants more discussion and research. (Dialogue has begun with the coordinator of the General Education Program.)

This proposal has been generated by the belief that the fabric of critical thinking is woven with analysis and synthesis of information and can be threaded with various application software. It is the hope of the authors that this paper will spark (potentially many), ignite (probably some), engulf (possibly a few), and not burn out.

References

[1] Anderson,R., Chairman's Message, ACM SIGCUE Bulletin, Vol.17, No.2, Spring 1983, p.1.

[2] Abrahams,P., "What is Computer Science", President's Letter, Communications of the ACM, Vol.30, No.6, June 1987, pp.472-473.

[3] Halaris,A. and Sloan,L., "Towards a Definiton of Computer Literacy for the Liberal Arts Environment", ACM SIGCSE Bulletin, Vol.17, No.1, March 1985, pp.320-326.

[4] Dyck,V.,Black,J., and Fenton,S.,"Beyond Traditional Computer Literacy", ACM SIGCSE Bulletin, Vol.19, No.1, February 1987, pp.508-512.
[5] Bailey,M., "Spreadsheets and Databases - Alternatives to Programming for Non-Computer Science Majors", ACM SIGCSE Bulletin, Vol.19, No.1, February 1987, pp.499-503.
[6] Computing Sciences Accreditation Board, Inc., Criteria for Accrediting Programs in Cpmputer Science in the United States, January 23, 1987.
[7] Arons,A.,"Critical Thinking and the Baccalaureate Curriculum", Liberal Education, Summer 1985, pp.141-157.
[8] McMillen,L., "Many Professors Now Start at the Beginning by Teaching Their Students How to Think", Chronicles of Higher Education, No.32, March 1986, pp.23-5.
[9] Stuckey,J., "The Computing Skills Workshop at Carnegie -Mellon University", ACM SIGCUE Outlook, Spring/Summer 1986, pp.46-48.
[10] Isaksen S. and Parnes,S., "Curriculum Planning for Creative Thinking and Problem Solving", The Journal of Creative Behavior, No.19, 1985, pp.1-29.
[11] Sternberg,R., "Teaching Critical Thinking, Part 1: Are We Making Critical Mistakes?", PHI DELTA KAPPAN, November 1985, pp.194-198.
[12] Sternberg,R., "Teaching Critical Thinking, Part 2: Possible Solutions", PHI DELTA KAPPAN, December 1985, pp.277-280.
[13] Mayer,R.E.,Dyck,J.L., and Vilberg,W.,"Learning To Program and Learning To Think: What's the Connection?", Communications of the ACM, Vol.29, No.7, July 1986, pp.605-610.
[14] Orndorff,J., "Computer Aids for the Teaching of Critical Reading in the Liberal Arts Curriculum", Proceedings of National Educational Computer Conference June 1987, p .208-212.
[15] Parker,J., ".ools for Thought", Journal of the International Council for Computers in Education, The Computing Teacher Conference Issue, 1987, pp.11-12.
[16] Parker,J.D. and Schneider, G.M., "Problems With and Proposals for Service Courses in Computer Science", ACM SIGCSE Bulletin, Vol.19, No.1, February 1987, pp.423-427.
[17] Bowers,C.A., "Teaching a 19th Century Mode of Thinking Through a 20th Century Machine", Proceedings of the National Educational Computer Conference, June 1987, pp.114-118.
[18] Kean College of New Jersey, "CPS 1032 Course Outline: Principles of Information Processing Using Microcomputers", February 1987.
[19] Slavin,R.E., "Learning Together", American Educator, Vol.10, No.2, Summer 1986, pp.6-13.

[20] Levine,M., "Docemur Docendo (He Who Teaches, Learns)", American Educator,Vol.10, No.3, Fall 1986, pp.22-25,48.
[21] Spence,J. and Windsor,J., "Establishing and Managing a Microcomputer Laboratory - Part III", Interface: The Computer Education Quarterly,Vol.8, Issue 4, Winter 1986-87, pp.12-17.