The Consortium for Computing
In Small Colleges
presents
the

Proceedings of the Seventh Annual

Small College Computing Conference

April 19-20, 1996
Midwestern State University
Wichita Falls, Texas

Edited by
Robert B. Sloger
Jennifer R. Pickle
Amarillo College, Amarillo, Texas

The Journal of Computing in Small Colleges

The Newsletter of the Consortium for Computing in Small Colleges

Volume 11, Number 7 **May 1996**

COMPUTER LITERACY ACROSS CURRICULUM

Marcin Paprzycki
University of Texas of the Permian Basin
Mathematics and Computer Science
4901 E. University
Odessa, Texas 79762

ABSTRACT
It is a general requirement for a college graduate to be computer literate. To achieve this goal various Computer Literacy courses are offered. It will be argued that passing a typical course does not make a person computer literate. An alternative solution that hopefully makes a person truly computer literate is offered.

1. COMPUTER LITERACY IN CURRENT EDUCATIONAL PRACTICE

The most recent set of accreditation requirements by SACS (Southern Association of Colleges and Schools) specifies that:

> The institution must demonstrate that its graduates are competent in reading, writing, oral communication, fundamental mathematical skills *and the basic use of computers* (SACS, 1995, p. 26, emphasis added)

The phrase "the basic use of computers" admits of various interpretations. It is predictable that the most typical way in which the universities and colleges belonging to SACS will deal with this requirement, will be through requiring students to complete a Computer Literacy course. Let us therefore investigate the contents of Computer Literacy courses currently offered to see what areas of the basic computer use are emphasized (for additional information pertinent to the subject see the extended list of references at the end of this paper). Recently the outcomes of a survey-based data collection related to the content of Computer Literacy courses were presented in Mitchell and Paprzycki (1992, 1993). Although the information was gathered with the primary goal of assessing the computer literacy requirements for prospective teachers it can be assumed that the results can be naturally extended to the whole student population. The primary finding was three basic types of courses are offered which when passed are claimed to make people computer literate:

(a) a high level programming language course,

(b) a course designed around the use of word processing, spreadsheet and database software,

(c) a course designed to evaluate, use, and incorporate software in the student's discipline.

It can be argued that these results suggest the existence of at least three different ways of understanding the term "computer literacy" (a more extensive discussion of this suggestion can be found in Mitchell and Paprzycki, 1992). A person is considered computer literate if she/he:

either
(i) is able to do simple computer programming,
or
(ii) is capable of using basic software packages,
or
(iii) can use the computer as a tool in a workplace.

The notion of computer literacy (i) (and courses similar to (a) when used to advance computer literacy) is clearly a relict from the times when the *only* way to use a computer was by writing programs. Then the *only* notion of computer literacy could have been the knowledge of computer programming. It is sometimes argued that today when using Visual Basic (or the macro language of one of the application packages) the elements of programming are still involved. This is not a very strong argument as in reality only very few computer users will be forced to program (even in this weak sense). In addition, the current development of computer environments is clearly geared toward reducing the need for programming skills on the part of the users (for instance through the introduction of intelligent helping agents — see, for instance, recent products from Microsoft). Although a person who knows how to program is certainly computer literate, computer programming is not the type of basic computer knowledge that SACS expects graduates to have.

The notion of computer literacy (ii) would be most likely supported by SACS as the one that provides the student with basic computer skills. At the same time this understanding of computer literacy seems to be somewhat outdated. It was relatively useful at the times when there existed a limited number of basic software packages (a word-processor, a spreadsheet, a database management program) but nowadays it seems much too narrow to be acceptable. Taking into account how fast software changes and diversifies, knowledge that may have been useful a couple of years ago may be entirely useless today. For instance, in the early 1980's there existed students who gained an in depth knowledge of WordStar. Such a knowledge would be totally useless today as (most likely) no WordStar word processors are currently in use. (A similar situation might be occurring before our very eyes with Word Perfect which may be all but extinct relatively soon.)

The final notion of computer literacy (iii) would be also most likely supported by the SACS as satisfying their basic skills criteria, but from the students point of view it is not without its flaws. This is so primarily because young people in early years of their careers change jobs relatively often. This does also mean that they may change the occupation. In this case it is difficult to imagine how the preparation in one area will be particularly useful in other areas.

Summarizing, the SACS accreditation requirements can be easily satisfied by requiring students to take most of the typical Computer Literacy courses. At the same time most of these courses are based on a notion of computer literacy that is outdated, too narrow and most likely does not provide students with the right kind of basic computer skills for their professional future. We would like to suggest that instead of starting from the SACS requirements we should start from an appropriate notion of computer literacy. Then we should see how this definition can be satisfied and finally see whether the SACS requirements would be thereby satisfied. Out of the "definitions" of computer literacy that were specified above the last one (iii) seems to be pointing in the right direction and we will use its generalized version. We will thus define literacy as an ability to use computers in all life-situations (at home, at work and others). More precisely, we will accept the definition proposed by Duckett (1992): *To be computer literate a person must have comprehensive skills, knowledge and understanding of computers and their use as they relate to technical, social and educational issues of the day and in the future.*

2. COMPUTER LITERACY COURSES VS. COMPUTER LITERACY

Bearing the above notion of computer literacy in mind let us examine whether the courses described in (b) and (c) above lead students to become computer literate. It is quite easy to see that the most typical Computer Literacy courses that introduce students to word-processing, spreadsheets and database software, do not achieve the goal of making them computer literate (even if we neglect for a while the objection that the knowledge gained is temporarily constrained). The most useful knowledge (and the only one that should be considered as being directly related to computer literacy) is how to use a word processor. (This is because word processors seem to replace handwriting and typewriters in the everyday life at a very high pace.) By contrast, spreadsheets are a very specific tool and their usage has a relatively limited application. It can be assumed that if the student will be ever asked to balance her check book she would rather use Quicken (or whatever will be the most popular money management software at the time). If a Psychology major will be (some time in the future) required to do basic statistical analysis of some data he will use an appropriate statistical package (which will most likely have a very nice psychology-oriented interface) instead of trying to develop statistical package in a spreadsheet (by writing programs in a given spreadsheet macro-language). The most difficult to accept as advancing computer literacy is an introduction to database software. It is very difficult to imagine that someone would use

database management package to develop a database of her CD's. What should be expected is that she will use a database-type software (with state of the art CD-related interface) to organize her collection. Summarizing, although a typical Computer Literacy course (in category (b)) definitely introduces students to computers, it does not seem to make them computer literate in the sense defined above.

Let us now analyze application-area-oriented Computer Literacy courses (c). The primary objection to these courses is that a typical student needs to be in some sense "broken into" computers first to be able to gain any real knowledge from (and about) them. It is really difficult to imagine that someone who has never used a computer can gain a lot of knowledge from a course that investigates e.g. the methodological issues of using various software packages in a classroom setting. Such a person will get so frustrated at the beginning that his chances of finishing the course are minimal. In addition, the knowledge gained will be rather narrowly focused thus defeating the purpose of making the student computer literate. Summarizing, the application-area-oriented courses possibly provide the right kind of knowledge, but at the wrong level (of breadth as well as depth).

Are we thus forced to accept a need for a two course sequence where the first will be an improved version of the standard Computer Literacy course (for instance by replacing the database part with an Internet oriented part) and the second will be area oriented Computer Literacy course? (This suggestion was made in the Mitchell and Paprzycki papers, but it will be rejected here.) Or, should we assume that quite soon all the students will have access to computers at home (or at school), so they will be computer literate entering college? It will be argued that another solution is possible that avoids most of the problems described above and makes graduating students computer literate *with no need to take Computer Literacy courses at all.* In addition this approach will also satisfy the SACS accreditation requirements.

3. COMPUTER LITERACY WITHOUT A SEPARATE COURSE

The suggestion is very simple. Instead of introducing a *Computer Literacy course* let us introduce *computers across the curriculum*. Every course a student takes should force her/him to use computers in one way or another. Let us first sketch how computer usage could be introduced in the courses that belong to the General Education Core Curriculum at University of Texas of the Permian Basin (these courses must be taken by every student). It should be pointed out that this is only a general outline and the details (which may vary depending on the particular course) are not too important at this stage. What is crucial is that in every course taken a computer component will be a part of the coursework.

(A) English (9 credit hours), History (6 credit hours), Government (6 credit hours)

- introduction to word processing (at UTPB MS Word) — including basic editing, saving and restoring files, cut/paste, basic page layout, footnotes and endnotes,
- introduction to library resources — electronic catalogs, database searches, microfilms, CD-ROM's, Internet-based materials,
- introduction to basic Internet services: e-mail, file transfer,
- introduction to Listserv based literature, history and government related discussion groups,
- introduction to Usenet based information services and discussion groups.

Since these courses are usually among first to be taken and since the topics covered overlap some coordination across the university is necessary (who will introduce what). The most important points are: *all* homework in the above described courses are accepted only if presented in a word-processor prepared form. The participation in at least one Listserv-based discussion group as well as proof of usage of Usenet-based information resources is required and possibly graded. E-mail based communication with the faculty members is encouraged. It is assumed that the computer knowledge gained in these courses will be used as a base for the computer usage in the remaining courses in the General Education Core. At the same time knowledge of the computer-related material from any one of these courses should not be treated as a formal prerequisite for other courses.

(B) Mathematics (3 credit hours) — introduction to a mathematics related software package (e.g. Derive, Mathcad, Maple),

(C) Mathematics, Statistics or Logic (3 credit hours) — introduction to additional mathematics/statistics/logic related packages and/or more extensive use of software introduced in (B).

This requirement is not exactly a new one. It is a well established practice to use technology in Mathematics education (see Paprzycki and Vidakovic, 1993, for an overview and bibliography). In many Statistics courses some statistical software is used (SPSS, SAS). Recently books have been published that introduce usage of computers to Logic courses (see for instance, Portoraro and Tully 1994). Since the mathematics-related parts of standard word-processors are not very sophisticated or easy to use, the mathematics areas may be the only ones where typewritten homework is not required.

(D) Speech (3 credit hours) — introduction to presentation software and hardware.

It is clear that the professional world is moving toward using presentation tools as a standard. It is thus extremely important that students be exposed to such tools. It seems accordingly sensible to include making professional presentations not only in a Speech course but also in other classes (an example of how professional presentations can be introduced into the CS curriculum can be found in Paprzycki and Zalewski, 1995).

(E) Visual and Performing Arts (3 credit hours) — introduction to graphics and painting packages, hypermedia and similar software.

Currently a computer-based Print-Making course is offered by the Art Department at UTPB. (This course is based on the Superpaint software package and the Quicktake camera and scanner.) The course is very well received and definitely generated a lot of interest among art majors.

(F) Social Sciences (3 credit hours) — usage of Usenet based discussion groups and news services, possible usage of statistical software, experiments with IRC, MUDD, MOO groups.

A word of caution is in place. During the Fall 1994 semester a MUDD epidemic was observed at UTPB. At the beginning of the semester two students introduced the "game" to their colleagues. By the end of the semester a large number of students was involved in MUDD'ing and approximately ten of them were playing non-stop during the time when the computer laboratory was open. Although this in itself is a very interesting phenomena that might be studied in certain Social Science courses, it can also be extremely addictive.

(G) Natural Science with a laboratory component (8 credit hours) — introduction to software packages appropriate to the discipline (for instance in Physics there exist a variety of laboratory experiments software that allow simulating experiments with laboratory equipment that may not be available at a given school; spreadsheets are very popular in the Chemistry courses to facilitate basic chemical calculations and rudimentary data analysis).

It is clear that a student finishing the core curriculum courses will have a very broad knowledge of computer usability in a variety of disciplines. The computer-related content of the courses presented above was selected in such a way that they can be taken in almost any sequence and that most of the applications do not overlap too much between the courses (except for word processing) — thus further increasing the breadth of knowledge. Since the

above suggestion is merely a sketch, it goes without saying that the material would have to be modified individually by each college and university.

It should be stressed that each discipline would introduce computers only in a "limited" capacity. It can be estimated that no more than 2-3 hours total (per course) would be used to introduce the software and its various features. An incremental approach would be used here, where during a particular class period only one or two new features of a software package would be introduced (taking approximately 5-10 minutes of a class period). The initial introduction to computer hardware (how to turn it on/off, where to insert a diskette, how to copy files) should be conducted by the Computer Center in a (free of charge) 1-hour introductory workshop. It should be pointed out that a limited number of sections of standard Computer Literacy could be left among the course offerings for the students who consider themselves computer-phobic. It is also obvious that if the approach described above is accepted, the SACS requirements are also satisfied as more than just a basic computer competency is achieved.

More extensive computer related activities should be introduced also in the remaining (mostly upper level) courses leading to the academic major and minor (as well as, possibly, teacher certification). This part of the computer education would match the computer literacy notions included in the currently existing courses of type (c). They should provide students with a broad knowledge about computer usage in their appropriate disciplines. At the same time the above mentioned problem with students who are completely computer illiterate and thus cannot be exposed to the more advanced topics would be removed. Typically, students take courses that belong to the core curriculum first and the courses on advanced topics later so they will be computer ready by the time the more advanced topics would be encountered. In addition the core curriculum computer usage provides students with knowledge broad enough to support their possible occupation changes.

Finally, observe that if this proposal is accepted and there was a Computer Literacy course required from all the students, it can be now replaced by a different requirement. This may be a very appealing perspective to everyone who feels that our graduates are underprepared when graduating.

4. CONCLUSION

If the above mentioned curriculum innovation is accepted it can be claimed that a graduating student is computer literate without having taken a separate course to achieve this goal, and the college/university has satisfied the SACS accreditation requirements in the area of computer skills. After graduation, students should be able to interact will all kinds of software and hardware and should not be surprised when confronted with a new software tool. For those students this will be just another software tool which can be easily mastered

because they have already seen many of them (similar and different ones). This is precisely how the goal *of having comprehensive skills and knowledge and understanding* will be achieved. At the same time it is obvious that the proposed innovation will not be easy to implement as it will require that all of the faculty members are computer literate and capable of using computers in their own discipline.

REFERENCES

D. Aiken (1992) An Integrated Approach to Computer Science, Mathematics and Problem Solving for the Liberal Arts Major, *Journal of Computing in Small Colleges*, 8(2), 146-153

S. Armstrong (1995) Computer Literacy -- the Well Traveled Road, *Journal of Computing in Small Colleges*, 10(5), 152-156

G. Atkins, M. Baugher (1995) The General Education Course: a New Perspective, *Journal of Computing in Small Colleges*, 10(5), 67-76

C. Bareiss (1995) The Future of Computer Literacy, *Journal of Computing in Small Colleges*, 10(4), 8-11

R. Bryant, P. De Palma (1995) Computer Science for Poets (and Other Smart People), *Journal of Computing in Small Colleges*, 11(2), 114-118

G. Duckett (1992) Basic Problems in the Development of a Computer Literacy Curriculum, *ACEC'92 Computing the Clever Country?*, Computer Education Group of Victoria, Richmond North, Australia

G. Gagne, B. Othmer (1994) Evolution of the Computer Literacy Course at Westminister College, *Journal of Computing in Small Colleges*, 9(4), 129-133

G.K. Gupta (1992) Computer Literacy at Small Liberal Arts Colleges, *Journal of Computing in Small Colleges*, 7(3), 122-128

C.A. Hall, H.W. Means, R.R. Sloger, D. Payne, (1992) The Changing Face of the Service Course, *Journal of Computing in Small Colleges*, 7(5), 47-49

B. King (1993) Taking the "Science" out of Computer Science, *Journal of Computing in Small Colleges*, 8(3), 6-12

J.S. McKeown, L. Molstad (1993) Creating a Computer Literate Campus, *Journal of Computing in Small Colleges*, 8(4), 34-38

H. W. Means (1991) Using Literature in a Computer Science Service Course: Improving Abstract/Critical Thinking Skills. *Journal of Computing in Small Colleges*, 6(5), 26-29

H.W. Means, S.I. Means, (1992) Incorporating Critical Thinking Skills in the Service Course, *Journal of Computing in Small Colleges*, 7(3), 116-121

T. Mitchell, M. Paprzyckl (1992) Teaching New Technologies: Developing a Computer Literacy Course for Prospective Teachers, *Journal of Computing in Small Colleges*, 8(3), 65-70

T. Mitchell, M. Paprzyckl (1993) Using Technology in the Classroom: Today and Tomorrow, *Technology and Teacher Education Annual 1993*, AACE, Charlottesville, 219-223

W. Mitchell (1993) Finding a New Direction for Computer Literacy, *Journal of Computing in Small Colleges*, 8(4), 1-9

D. Novak (1992) Art and Problem Solving Through a Computer Literacy Course, *Journal of Computing in Small Colleges*, 8(2), 177-182

R. O'Lander (1992) A Multicultural Approach to Computer Literacy, *Journal of Computing in Small Colleges*, 7(3), 133-137

M. Paprzyckl, J. Zalewski (1995) Should Computer Science Majors Know How to Write and Speak?, *Journal of Computing in Small Colleges*, 10(5), 142-151

M. Paprzyckl, D. Vidakovic (1993) Using Computers in Calculus Teaching, *Journal of Computing in Small Colleges*, 8(5), 34-45

G. Pitts, J. Howland (1994) Computer Skills: an Essential University Requirement, *Journal of Computing in Small Colleges*, 9(5), 34-40

F. D. Portoraro, R. E. Tully (1994) *LOGIC WITH SYMLOG Learning Symbolic Logic by Computer*, Prentice Hall, Englewood Cliffs

J. Rabung (1993) Real Computer Concepts in a Literacy-Type Setting, *Journal of Computing in Small Colleges*, 8(3), 90-93

SACS (1995) Criteria for Accreditation, Southern Association of Colleges and Schools Publication.

K. Smith (1993) Enjoying a Computer Literacy Course: Discussion of Critical Thinking Skills, *Journal of Computing in Small Colleges*, 8(3), 1-5

R. L. Tureman Jr. (1992) Infusion of Writing in the CIS Curriculum Using Student Communication Skills to Promote Critical Thinking, *Journal of Computing in Small Colleges*, 7(3), 129-132

R. Von Holzen (1995) Effective Group Size in a Computer Literacy Class: Impact on Students' Computer Knowledge and Attitude, *Journal of Computing in Small Colleges*, 10(5), 77-83