SOME NOTES ON AN APPROACH TO TEACH(ING) COMPUTERS

for Eric & Lew from bp
November 14th 1985

How it strikes me is that there are two approaches one can take to using programming languages on a computer. I'm going to borrow from the language of poetics and refer to them as Open Programming and Closed Programming. Closed Programming is where you know what you want the computer to produce, you know the end product in mind, and therfore your approach is a structured one (as, say, in C programming). Open Programming is where you have no end product in mind but are proceeding step by step just to see where each little step takes you (as in, obviously, what lots of people do with the various versions of Basic). These are not hierarchical categories. One approach is not, in that sense, uncategorically better than the other. Circumstance and need determine which approach one uses and which approach is, therefore, most useful to the user.

Obviously this comes out of my own fiddling with poetry for the text screen. I have, in fact, invoked a degree of closure in my little programmes by refusing to use the graphics screen. But that's probably better described as a voluntary restraint i have chosen to work within. The point is that it seems more useful to me to approach the teaching of programming from an inclusive rather than an exclusive point of view i.e. why shouldn't one be able to go back & forth between open & closed programming approaches. From what i've seen that's in fact what most of even the most fanatic structuralists do (i.e. they have to play with certain commands to see what effects they have before they can incorporate them into their structures) & is certainly part & parcel of the notion of hacking. If one recognizes these as viable alternative methods then the notion that one is learning bad programming habits disappears. one is simply learning different approaches and each approach has its strengths and weaknesses.

Having said all that the next part of my proposal is probably clear. The notion is to devise a kind of beginning programming textbook, and/or a beginner's course in programming, which would utilize the two approaches, both the structured and unstructured, approaches and encourage experimentation in both approaches. Part of the strength of such an approach is that by utilizing things like my little visual poems (most of which could also be viewed as illustrations of applications) we could have something which opened up access for students of literature and art as well as students of math and the other sciences. The general orientation in the teaching of computers i've seen is essentially towards business type applications (which are, of course, incredibly valuable) or games (which are fun) but there are other grounds that would give both students and teachers additional avenues of approach and maybe make the way they're taught more flexible. These thots are based on admittedly limited experience but struck me as at least worth talking about to you two.