

American programming's plight.

A competent programmer's most important assets are - perhaps in this order - an excellent mastery of his native tongue and a considerable mathematical maturity. These facts are well-known and well-understood for more than ten years.

By a sad accident of history they make programming, which is difficult anyhow, in the USA even more difficult. Firstly, since the turn of the century the American culture has gradually turned itself away from mathematics (see Lynn A. Steen "Mathematics Today" or Morris Kline "Mathematics in Western Culture") to the extent that many of its otherwise well-educated citizens have no longer a balanced view of what mathematics is all about. The culture has become amathematical, sometimes even antimathematical. Who could foresee in 1900 that by the end of the century, thanks to the advent of computers, it would be more rewarding than ever to have a mathematically trained mind? Secondly, (see, for instance, Edwin Newman "Strictly Speaking") the ideals of the permissive society have accelerated the deterioration of the average usage of the English language. (Other Western countries suffer from such a phenomenon too, but the USA

-as usual!- seems to be ahead of them.) For American programming, these two developments are very unfortunate indeed.

There is a further encumbering circumstance; I knew it, but a recent visit to some U.S. industries made me perceive it in its full seriousness. The trouble is that programming became an industrial activity at a moment (see, for instance, William H. Whyte "The Organization Man") that the prevailing management philosophy aimed at making companies as independent as possible of the competence of their employees. Needless to say, this gave American industrial programming a very false start. How strong this attitude still is I learned during my recent visit when a horrified American manager reacted upon a suggestion of mine with: "But that would require people to think!". It was as if I had made an indecent proposal. (The very common reaction to look immediately for "a tool" as soon as a problem emerges could very well be a symptom of that same attitude.)

One further circumstance blurs people's vision. For those who understand what programming is about and agree with Morris Kline's verdict "More than anything else mathematics is a method." it is obvious that programming must be

regarded as a branch of mathematics, be it a somewhat curious one. Certainly now, but perhaps for ever, programming distinguishes itself from most other branches of mathematics by a different balance between the required knowledge and the required maturity. Programming requires relatively little mathematical knowledge; avoiding unmastered complexity, however, requires a great methodological awareness. I know that this unusual balance prevents some of my Dutch colleagues from viewing programming as a promising area of mathematical activity. More knowledge oriented than I, they keep asking "But where is the relevant mathematics?". They feel that the subject is "too shallow" or "not rich enough" to become a flourishing branch of mathematics. My recent visit to the USA reminded me of the fact that the American education is much more strongly knowledge oriented than the traditional European one. (A discussion about computing science education was immediately phrased in terms of the question "which subjects the students should take"; trying to improve their thinking habits wasn't considered at all.) The relatively modest rôle of mathematical knowledge in programming blurs the European vision; I am afraid it blurs the American one even more.

Plataanstraat 5
5671 AL NUENEN
The Netherlands

19 September 1980
prof. dr. Edsger W. Dijkstra
Burroughs Research Fellow.