

# Copyright Notice

The following manuscript

EWD 512: Comments at a Symposium

is held in copyright by Springer-Verlag New York.

The manuscript was published as pages 161–164 of

Edsger W. Dijkstra, *Selected Writings on Computing: A Personal Perspective*,  
Springer-Verlag, 1982. ISBN 0-387-90652-5.

**Reproduced with permission from Springer-Verlag New York.  
Any further reproduction is strictly prohibited.**

Comments at a Symposium

Ladies and Gentlemen,

Before airing a number of comments and remarks I would like to tell you something about my past lest I be misunderstood.

Firstly - and this is apparently in contrast to a number of people present - I consider myself as being a very lucky person because I am perfectly happy with the role that mathematics have played in my life. Extended over a period of 45 years, my mathematical education has been, I guess, about a 10 man year effort; you may not like the result, but I liked most of the experience immensely and that amount of fun and intellectual excitement I regard, all by itself, as a sufficient justification. Besides that, my enjoyment was untampered by the now fashionable quibble about "heredity" versus "environment", because in any case my dear mother played a major role in it.

Why do I bring this up? Well, simply, because the only fruitful way of considering computers, that I know of, is regarding them as mathematical machines. Knowing that, I came to this symposium with very low expectations, because this year's motto is "Computers and the educated individual". But mathematics, however, are no longer regarded as an essential ingredient of the cultural baggage of the educated man! Read Eric Temple Bell complaining about the watering down of the American high school, where mediocrity has become the norm, a degradation covered by a misuse of the notion "democratic". Read Courant's introduction to Morris Kline's "Mathematics and the Western Culture", and look around yourself: you will find many in your environment who pose as educated persons and simultaneously announce with some curious pride that "of course they never understood mathematics". Two generations ago, the pitiful one who found mathematics beyond him, tried to cover up his mental infirmity. In short: with today's "educated" individual, and with computers being mathematical machines, our subject "Computers and the educated individual" has a hard time in finding an area of application.

## 2.

To make things worse, the "educated individual" is so unfashionable as to have become nearly extinct. In the name of justice and equality, the bright pupils are no longer allowed to understand what the stupid ones cannot grasp, and many a government threatens the race of the well-educated individuals with genocide. In the hands of the pedagogues education has been replaced by training, what used to be sowing the seeds of understanding with a hope for harvest has been replaced by educational engineering. Even the individual had better disappear and submerge into a team as quickly as possible. Instead of "Computers and the educated individual", I propose the more appropriate title: "Computers and the ill-trained mob".

In that setting I have been asked about software in the next 25 years! The safest weather prediction for tomorrow is as we all know, "the same weather as today" and if I followed that line I should predict another 25 years of FORTRAN and COBOL. I expect this prediction to be true to a large extent, because there have always been enough fools in this world. But it is the kind of uninteresting prediction saying that tomorrow morning the sun can be expected to rise again. It would only be the full truth if the name of our subject were "stagnation".

Mind you, the pressures to enforce stagnation are strong enough. Sound financial principles seem to dictate that the more expensive our mistake the longer we must maintain it, and there are computing scientists that honestly believe that OS/360 is here to stay, from now until eternity, the argument being that it is too expensive to replace it. There is the possibility that we learn to make a better system at lesser cost; there is the certainty that it becomes too risky and too expensive to continue to use it. Already many a large organisation is nearly crushed under the sheer weight of the illogical, unmastered complexity of its automatic data processing systems. Things have to change and, therefore, will change. Perhaps we have to wait for a few more spectacular collapses until it dawns upon mankind that we had better understand what we are

## 3.

doing. I don't believe in stagnation, I do believe in patience. The current tools will be replaced by better ones because the current ones are just too inadequate.

Please do not misinterpret my appreciation of FORTRAN: if there had been a Nobel prize for computing science FORTRAN would have been an achievement worthy of it. But that appreciation should not engender the mistaken belief that FORTRAN is the last word in computing; on the contrary, it was one of the first words. It is just no longer adequate: since the twenty years of its existence, the computing scene has changed by several orders of magnitude. How could it still be adequate? We don't control Jumbo Jets by whip and spur!

There are two views of programming. In the old view it is regarded as the purpose of our programs to instruct our machines; in the new one it will be the purpose of our machines to execute our programs. In the old view a programmer's expertise is proportional to his knowledge of all the funny properties of the equipment against which he has to fight a continuous battle. In the new view a programmer's competence is displayed by his good taste and the justification with which he rejects inelegant implementation and clumsy interfaces. In the old view, programming becomes easier when the machines become faster and bigger because we can then stay away further from the limits of their capacity; in the new view (recognising that before we had machines, programming was no problem at all), it is recognised that our programming problems will grow with the power of our machines, because we will become more ambitious.

I am perfectly convinced that there will come a time when it will be recognised that programming is one of the more difficult branches of applied mathematics because it is also one of the more difficult branches of engineering, and vice versa. I am equally convinced that simultaneously programming will evolve from a craft learned by apprenticeship, into an intellectual discipline that can be taught and studied, and need no

## 4.

longer be based on the technical mistakes of the department of defence and the computer manufacturers. Don't blame me for the fact that competent programming, as I view it as an intellectual possibility, will be too difficult for "the average programmer" - you must not fall into the trap of rejecting a surgical technique because it is beyond the capabilities of the barber in his shop around the corner.

To imagine the teaching of a discipline of programming as a science requires some imagination. Any effort to teach programming while disguising its intrinsic mathematical nature, is doomed to failure, but we shall have to teach a discipline of programming in a way that differs from the average way in which mathematics is taught today. The problem with today's mathematical curricula is that mathematical results are published and taught quite openly, but how mathematics is done is not published, not taught explicitly, and the student must pick it up by osmosis so to speak. In this respect mathematics is only half-way between the open science and the secret craft of the guilds, and we are forced to observe that the great majority of trained mathematicians are only amateur thinkers.

But programming, when stripped of all its circumstantial irrelevancies, boils down to no more and no less than very effective thinking so as to avoid unmastered complexity, to very vigorous separation of your many different concerns.

As far as my experience goes programming in the sense of thinking, or thinking in the sense of programming, can indeed be taught. Not all of your students will learn it, but in that respect it is no different from any other subject. Polya's "How to solve it" and his "Art of Plausible Reasoning", although inspiring, are not enough. That would be more than can be expected, for the programming problem only emerged after those books by Polya had been written. And perhaps Polya tried to teach something more elusive than what we are trying to teach now. Polya was

## 5.

concerned with problem solving and he made a sort of checklist that one could go through when trying not to overlook the in some sense "surprising" or "unexpected" solution. But this time we are not so much concerned with problem solving in Polya's sense. I think that "solution composition" comes much closer to what we have to do now. We have to fight chaos, and the most effective way of doing that is the prevention of its emergence. We have to learn to avoid all forms of combinatorial complexity generators that when active rapidly tax our ability to carry out a case-analysis far beyond the limits of our power of reasoning. To recognise the emergence of a combinatorial complexity generator long before it has poisoned your design beyond salvation requires constant vigilance, a vigilance that can and should be taught. To circumvent such emerging complexity generators may very well be a tough problem the solution of which I can only describe as mathematical invention. A great advantage is that we know at least what we are looking for, and - perhaps most important of all - that a terminology is emerging with which we can name the different stages and aspects of our intellectual endeavour, a terminology in which we can answer the otherwise frustrating question that so often emerges in the midst of one's struggles in "What the hell am I really doing?".

The main virtue of machines is that they have confronted us with a new class of extremely difficult problems that, with love, luck and discipline, we shall learn to cope with. As a reaction to this challenge, consciously trained thinkers will emerge: we need them. The first consciously trained thinkers will be largely self-taught ones, but ..... consciously trained, and they will learn how to educate others.

No one needs to tell me that with all its political and social implications this will be a very slow process, much slower than technically necessary. It is that "ritenuto" enforced by society that may see to it that my prediction is good for - as Ewan Page asked - the next 25 years.