

To the members of the Budget Council

I write to you because of a rumour of efforts to replace in the introductory programming course of our undergraduate curriculum the functional programming language Haskell by the imperative language Java, and because I think that in this case the Budget Council has to take responsibility lest the decision be taken at the wrong level.

You see, it is no minor matter. Colleagues from outside the state (still!) often wonder how I can survive in a place like Austin, Texas, automatically assuming that Texas's solid conservatism guarantees equally solid mediocrity. My usual answer is something like "Don't worry. The CS Department is quite an enlightened place, for instance for introductory programming we introduce our freshmen to Haskell"; they react first almost with disbelief, and then with envy --usually it turns out that their undergraduate curriculum has not recovered from the transition from Pascal to something like C++ or Java.

A very practical reason for preferring functional programming in a freshman course is that most students already have a certain familiarity with imperative programming. Facing them with the novelty of functional programming immediately drives home the message that there is more to programming than they thought. And quickly they will observe that functional programming elegantly admits solutions that are very hard (or impossible) to formulate with the programming vehicle of their high school days.

A fundamental reason for the preference is that functional programs are much more readily appreciated as mathematical objects than imperative ones, so that you can teach what rigorous reasoning about programs amounts to. The additional advantage of functional programming with "lazy evaluation" is that it provides an environment that discourages operational reasoning.

Finally, in the specific comparison of Haskell versus Java, Haskell, though not perfect, is of a quality that is several orders of magnitude higher than Java, which is a mess (and needed an extensive advertizing campaign and aggressive salesmanship for its commercial acceptance). It is bad enough that, on the whole, in-

dustry accepts designs of well-identified lousiness as "de facto" standards. Personally I think that the University should keep the healthier alternatives alive.

\* \* \*

It is not only the violin that shapes the violinist, we are all shaped by the the tools we train ourselves to use, and in this respect programming languages have a devious influence: they shape our thinking habits. This circumstance makes the choice of first programming language so important. One would like to use the introductory programming course as a means of creating a culture that can serve as a basis for a computing science curriculum, rather than be forced to start that with a lot of unlearning (if that is possible at all: what has become our past, forever remains so). The choice implies a grave responsibility towards our undergraduate students, and that is why it can not be left to a random chairman of something but has to be done by the Budget Council. This is not something that can be left to the civil servants or the politicians; here statesmen are needed.

Austin, 12 April 2001

Edsger W. Dijkstra