HOARE73

Hoare, C.A.R.; Hints on Programming Language Design; SIGACT/SIGPLAN Symposium on Principles of Programming Languages (October 1973).

The central thesis of this paper is "that the primary purpose of a programming language is to help the programmer in the practice of his art." Hoare suggests that this is often overlooked by language designers. In the first part of the paper, he lists what he considers the most difficult aspects of a programmers task and suggests how programming language design can assist in meeting these difficulties. He then attacks a number of the goals which had been followed up to that time. The rest of the paper is not very cohesive, consisting of a number of topics from arithmetic expressions to variables to block structure. In this section, Hoare takes the position that certain aspects of high-level languages are better than machine-level languages and certain ones are worse. The basic idea is that high-level languages should remove the dangers inherent in machine-level programming.

Hoare claims that program design, documentation, and debugging are the most difficult aspects of the art of programming. The ideas emphasized can be summarized as simplicity, security, fast translation, efficient object code, and readability. Each of these points is examined from the point of view of past work as well as suggestions for the future. He makes a strong argument against separate (independent) compilation as a method to speed up compilation, pointing out that wide interfaces generally exist between separate parts of the program and that even worse, the validity of these interfaces is not checked at compile time.

The underlying theme is that language design at the time had been headed in the wrong direction. Hoare is arguing for simpler, cleaner high-level languages with a few suggested new features. His final suggestion is that language designers should not design new features, but should rather make use of a subset of the existing ones. The subset should be chosen for the particular use intended for the language, with minor modifications made in the case of inconsistencies between desired features (i.e. choose a small set of orthogonal features, adjusting features taken from different languages so that they do not have overlapping meanings or applicability).