

p645
MR

2 Comparative Programming languages

2002

Briefly discuss the compromises that must be made when standardising a programming language. [8 marks]

Discuss the relative merits to (1) the application programmers and (2) compiler writer of the following ways of specifying a programming language.

- (a) A concise readable user manual for the language in english containing many useful programming examples.
- (b) A very long and highly detailed description, in english, of every feature of the language. This manual contains no programming examples.
- (c) A concise but rigorous description using a formal grammar to describe the language syntax and making extensive use of mathematical notations taken from set theory, λ -calculus, predicate calculus and logic to describe the semantics of the language.
- (d) The source code for a clean and elegant machine independent interpretive implementation of the language.

[12 marks]

ANSWER:

A) English can be easy to read but is rather imprecise. Formal notations are often not understood by ordinary users and are often hard to follow. When should the standard be written -- early v late. How much freedom should be given to the implementer -- order of evaluation, arithmetic precision, etc. Efficiency v machine independence.

B) Bookwork. Leaves plenty of room for discussion. a) Woolly imprecise, but readable and short. b) Probably hard to find what one wants to know but possibly ok for compiler writers. c) Fine for experts. d) Possibly good but tends to overspecify the language. Does not easily allow easy specification that eg operands of an expression can be evaluated in either order.