OCEAN DATA SYSTEMS, INC.
6000 EXECUTIVE BLVD., ROCKVILLE, MARYLAND 20852 • 301/881-3031

Feb. 27, 1974

Prof. D. Knuth.
Stanford University, Stanford Calif.

Dear Prof. Knuth.

Here is some of the material which I promised to look up and send along to you. All but one date from the '50's, and at least one is what I think Fred Gruenberger had in mind. In retrospect, there doesn't seem to be much here, but I know that I got awfully tired of trying to say the same things over and over and having them fall on deaf ears. The allures of ever-higher-level languages were strong, and structuring wasn't very appealing by comparison; besides, structuring is perforce restrictive, and we couldn't have that!

Unfortunately, many of the things which I believed (and still do) never got written down properly; my eloquence is such that I postpone such things, and end up doing more by example, so that my ideas are more apparent in code I've written & documented than in formal publication.

Still, the whole subject still fascinates me, and it's been great fun digging out this material and thinking about subjects that I haven't bothered with for years (I'm strictly applications (engineering) oriented these days.) In particular, I'm still fascinated by a particular idea: a great "break through" in computing (I still call it that) was the notion that a stored-program computer could operate not only on data, but on programs themselves. Since then, we've been trying to make sure that this never happens,

least of all at the programmer's behest. But certainly our software ought to be able to look at a program statement *qua* program element, and relate it to the whole, and not just treat it as an entity to be translated and incorporated into the whole. But this hasn't happened.

I suspect that this comes about as the result of our arbitrary symbolic nomenclature, which manages to eliminate any information about the thing named from the name of the thing; this doesn't happen much in other fields, which normally use highly structured naming systems as part of the basis of discourse. (Trivial examples: Elist Street; Room 501; aft guidance gyro; pin 5 of socket X3, etc). But when you start off with the premise that any arbitrary name can be used for anything, you are already on the way to trouble. And worse: the name may imply that the named entity has the properties only arbitrarily incorporated in the name (Examples: a subroutine named FINDROOT may or not find a root; the variable TWOPI may or not contain a value for $2\pi$ (what form, format, etc.?); the expression NETPAY EQUALS GROSSPAY MINUS DEDUCTIONS may or may not refer to any dollar values). But I wonder: as long as naming and structure are arbitrary, we can't get much help from system software. Contrary examples do exist, to a degree: the Douglas system had some of these features; so does JOSS; so do most special-purpose application-oriented interpretive systems.

Cliff Shaw (also ex RAND) and I once wrote down a list of concepts which we felt were essential to the

understanding of programming; we then amused ourselves by verifying that no papers being written on programming ever took cognizance of these. I have looked for the list but can't locate it. Context, I know was there, as was Denotation Designation, Hierarchy, Nomenclature, Delegation, Implication, Explication, etc. etc. Perhaps Structure and Form were there, too. Oh, well. Representation. Objects of Discourse. Homomorphism.

A final thought. I personally stand in awe of "The Art of Computer Programming". I have no conception whatever of how such a work comes into being! My highest praise is reserved for your fine sense of humor; keep that up at all costs. I'm hoping that eventually what 1960 ish RAND/Carnegie-ites call complex information processing will be treated. I don't know how to define this, but can give examples. One criterion is that the list structure representing the hierarchical sequencing of instructions is itself an object of discourse and manipulation by the language; so are the individual instructions and the collection of these into units. IPL-V had these features, but we used them in a "seat-of-the-pants" fashion. This subject needs scholarly discourse! Anyway, keep up the good work.

My apologies for long-windedness: I guess that's a result of reminiscences of the RAND days. Give My regards to Paul Armer, Ed Feigenbaum, etc.

Sincerely

Chuck Baker