29 [12].—CHARLES M. THATCHER & ANTHONY J. CAPUTO, *Digital Computer Programming: Logic and Language*, Addison-Wesley Publishing Co., Reading, Mass., 1967, xi + 159 pp., 28 cm. Price $3.95 paperbound.

This soft-cover book is one of the many to appear on the computer programming scene in recent months. It is significantly different from several of its competitors for various reasons.

In the first place, much time is devoted to a discussion of a hypothetical computer (a good idea in itself) called HICOP, an acronym for a Highly Imaginary Computer for Orientation Purposes. Unfortunately, the beginner is just as ignorant of the need for a HICOP as he is about the workings of any nonhypothetical computer. His task is not made easier by being thrown directly into a host of definitions and a technical discussion of flow-charts followed by an explanation of iterative looping.

Furthermore, the manner in which the material is presented is couched in the jargon of the sophisticate—hardly the language of the average person to whom the authors rightfully address themselves in the preface. Taking, almost at random, a paragraph out of the otherwise excellent preface, will serve to illustrate the reviewer's objections since this kind of technical mumbo-jumbo can only confuse and confound a beginner.

"Significantly, the hypothetical computer accepts the problem-oriented language directly as its own machine language; but subsequent modifications to the computer's capability make it more realistic in this respect. These modifications ultimately lead to a typical machine language, in terms of which the internal operation of real computers is summarized." The object of such a book as this surely should be the enlightenment of the reader.

In the introduction to indexing, the unwary reader is presented with the following gem:

"Any repetitive operation lends itself to iterative looping. It might therefore be anticipated that the iterative technique can be used to good advantage when many separate items of data are to be subjected to the same manipulation, even though no recurrence formula can be written, since the items are distinctly separate."

I know a good number of experienced programmers who would CALL EXIT at this kind of description.

HENRY MULLISH

New York University
Courant Institute of Mathematical Sciences
New York, New York 10012

30 [12].—GERALD M. WEINBERG, *PL/I Programming Primer*, McGraw-Hill, Inc., New York, 1966, ix + 278 pp., 24 cm. Price $5.95.

As stated in the Preface, this book is intended to be "an introduction to PL/I, to fill promptly the need for an introductory text on a new language. . . ." As such, it provides a fairly complete and readable introduction to the features of the language that it covers, but its value, even to the beginner, is severely diminished by omissions. Since this is one of the first books on the subject, and was written at a time (1966) when the PL/I language specifications were hazy and in a state of

flux, many of these omissions are understandable. However, the complete absence of at least two important PL/I statements (LOCATE and REVERT) and of numerous minor but important techniques, (e.g., indexing a structure BY NAME), prevent the reader from fully appreciating the power of the PL/I language.

The organization of the material, based upon the elaboration of a few basic programs to incorporate more advanced features, is highly commendable. It is far superior to the obviously contrived and trivial examples often used in textbooks to illustrate programming techniques. The exercises are adequate to illustrate and reinforce the major points of the book, and the answers provided seem to be correct and complete.

There is a conspicuous lack of the usual technical appendices (the only one provided is the PL/I character set). This is unfortunate since one would appreciate at least a list of legal statements and their formats. This is especially vital since several statements are never mentioned, or, if they are, they are not included in the index. Presumably, the reader is at the mercy of the IBM-provided language specification or Mr. Weinberg's promised "second volume" which "is contemplated for the time when the most advanced features of the language are more firmly specified." Perhaps this volume will continue in the excellent style of the first, and will provide a reasonably complete introduction to PL/I. In the meantime, there are several other texts on the market, notably Frank Bates and Mary Douglas, *Programming Language/One,* Prentice-Hall, Inc., Englewood Cliffs, N. J., 1967, which seems better fare for both the novice and the experienced programmer.

HENRY MULLISH

New York University
Courant Institute of Mathematical Sciences
New York, New York 10012

31 [12].—GUENTHER HINTZE, *Fundamentals of Digital Machine Computing,* Springer-Verlag, New York, 1966, ix + 225 pp., 24 cm. Price $6.40.

As pointed out in the preface, this book is "the outgrowth of class notes developed over several years for a basic course on digital computers. . . ". The approach is nearer that of the computer designer than that of the programmer, starting with the representation of numbers, and progressing through logical design, instruction codes, and assembly language to an introduction to automatic programming. The book is somewhat dated, showing the computer family tree ending at the IBM 7094 and the CDC 3600, and the most recent reference being the revised Algol report of January 1963. It contains a number of generalizations which may have been reasonably accurate several years ago, but which appear very curious in the light of recent developments. For example, the sentence "Typical timing figures for presently used circuits, coordinating pulses and voltage levels, are pulse width .25 microseconds, response time for flip-flop approximately 1.5 microseconds, and pulse intervals 2 microseconds" is in no way consistent with commercially available packages which are more than 10 times as fast, and machines like the CDC 6600 using flip-flops which switch in 20 nanoseconds.

The personal preference of this reviewer is for course material which contains factual information about machines or programs to which the student has access,