Book Reviews

HARRY KATZAN JR
Invitation to PASCAL
Van Nostrand Reinhold, New York, 1981.
233 + x pp. £13.15 (boards).

The programming language PASCAL was devised in 1968 and has become, particularly since the publication of the Revised Report[1] in 1973, increasingly popular in the Computer Science departments of Universities where such attributes as its fast compilation, good diagnostics, efficient object code and, above all, the portability of its compiler have been particularly appreciated, in addition to its more obviously computer-science oriented features such as the set and pointer types.

More recently, however, interest in Pascal has been awakened among non-computer scientists, including both converts from FORTRAN (who were quite unmoved by Algol 60 and its other derivatives) and converts from BASIC who want the greater power of Pascal; because of the latter group, Pascal is rapidly becoming *the* language for microcomputers. Not unexpectedly, there has been an accompanying boom in books on Pascal during the past two or three years, so much so that anyone now wishing to write a book on the subject should be expected to demonstrate an originality of approach or a superiority of quality in order to justify their efforts.

Regrettably, however, I could not recommend the volume now under review were it the only textbook on the subject. It is intended as the first of a three-volume set, to be followed by 'Intermediate Pascal' and 'Advanced Pascal', and covers a subset of the language, omitting records, sets and pointers as well as restricting itself to a subset of I/O facilities (i.e. the procedures read (ln) and write (ln) and the function eof applied to the standard files input and output). This much is commendable; but the book is poorly printed, uses non-standard terminology and contains a number of errors of fact which suggest a failure on the author's part to grasp some of the fundamentals of the language. A complete list of faults would be too long to include in this review, so I will include just a few examples of each kind of fault:

Poor printing

pp. viii–ix contain the text of pp. 111 (last three lines)–113 instead of the bulk of the Preface.

Table 2.1 (p. 14) contains two dashes (which are not part of the character set).

On p. 95 in items 2, 3 and 5 of the list of characteristics (of the collating sequence) we have capital O instead of zero (five times) and letter 'q' instead of figure '9' (twice).

In Exercise 5.1. (a) on p. 114 **DIV** is in too small a font.

In the text on p. 126 BEGIN and END should be in bold in common with other word-delimiters.

Throughout the text string quotes are printed as balanced pairs, although only the opening quote is shown as part of the character set—a practice which makes nonsense of the convention for including a quote in a string!

I found it irritating that not only the word 'PASCAL' but also each piece of program text appears in upper case instead of the much more legible and user-friendly lower case which the Report[1] uses. This practice has the marginal advantage that one can write (e.g.) 'the READ procedure starts at the current file position' without there being the same degree of ambiguity as were it all in lower case, but tends to propagate the myth that computers are semi-literates that can only read block capitals. I also found the use of a smaller font for the program fragments than for the body of the text confusing since the same identifier or word-delimiter appeared in the text in a larger font, and whereas I would have found the size of the text about right for normal (i.e. mainly lower-case) work, the over-use of upper case makes it look awkwardly large.

Non-standard terminology

Throughout the text the (**) comment delimiters are used instead of the { } of the Report.[2]

On p. 17 **VAR, BEGIN** and **END** are referred to as 'identifiers' although the Report[3] always calls them 'word-delimiters' (or 'reserved words').

On p. 38 and p. 45 the term 'procedure' is used when what is meant is 'block'.

The term 'enumerated type' is introduced on p. 40 as a synonym for 'scalar type', whereas what is meant is 'declared scalar type' of the Report.[4]

The term 'ordinal type' is introduced on p. 47 to embrace 'INTEGER, BOOLEAN, CHAR, enumerated and subrange types'. However, the Report[5] only uses the term 'unstructured type' to embrace all scalar and subrange types.

The term 'basic data type' is used on p. 110 (in connection with subrange types), whereas the Report[6] uses 'associated scalar type'.

The term 'line boundary character' is introduced on p. 67 instead of the 'line separator' of the Report.[7]

Errors of fact

On p. 26 '14.' is given as an example of a real number although this is not only at variance with the Report[8] but with the syntax diagram of Figure 2.5 (p. 24).

On p. 28 the semicolons are missing from the first compound statement.

In Table 5.3 (p. 92) the natural logarithm function is given the identifier 'LX' instead of the 'ln' of the Report.[9]

On p. 132 it is stated that

ARRAY [index] **OF**
  ARRAY [index] **OF**
    ...
      ARRAY [index] **OF** component

... can be abbreviated in PASCAL as:
ARRAY [index, index, ..., index] **OF** component'

but an array of arrays is quite different from a multi-dimensional array.

In particular, the former allows slicing, the latter does not.

On p. 152 there is a missing semicolon from the BNF syntax of the **REPEAT** statement.

On p. 188 it is stated that, since 'The **VAR** option may not be used in the formal parameter list of a function definition' side-effects are limited. This is wrong, since non-local variables may have their values changed by the call of a function.[10]

On p. 195 it is stated that the use of type identifiers to shorten the procedure heading is a matter of convenience. In face the Report[11] *requires* that type identifiers must be used in the procedure heading.

Finally, I did not find that the introduction of BNF at a late stage in the book added anything to the 'tramline' syntax diagrams used from the beginning, nor did the 'structograms' make the meaning of the control structures any clearer. Fortunately, the price is on the high side for the first of a three-volume set when there are many comprehensive one-volume works on this topic at a comparable price, or for much less in paperback form; I hope that this fact alone will prevent its becoming widely read. To parody learned counsel in the 'Lady Chatterley' case, 'Is this a book you would like your students to read?' For my part, the answer is an unequivocal 'No'.

DAVID J. CAIRNS
London

References

1. K. Jensen and N. Wirth, *Pascal User Manual and Report*, 2nd Edn, Springer-Verlag, New York (1975). ISBN 0-387-90144-2
2. ibid., 1 (p. 9).
3. loc. cit.
4. ibid., 2 (p. 12).
5. loc. cit.
6. ibid., 5.A (p. 34).
7. ibid., 9.A (p. 57).
8. ibid., D (p. 111).
9. ibid., A (p. 106).
10. ibid., 11.B (p. 79).
11. ibid., D (p. 117).

P. KATZ
Digital Control Using Microprocessors
Prentice Hall, Englewood Cliffs, New Jersey, 1981. 293 pp. £16.95.

Do not be misled by the title of this book—it is meant for the control engineer thinking of implementing a control strategy involving microprocessors and, as such, the emphasis is upon control theory with a small proportion given over the microprocessors.

Nevertheless, the book is very informative and, undoubtedly, a useful reference book for the more dedicated reader. Since the material for the book comes from a course on digital control at the Technion, Israel Institute of Technology to final-year engineers, it is well prepared with a couple of interesting design examples.

Though I enjoyed reviewing this book, it must be said that it is not a book to read from cover to cover, but only by chapters. Finally, I would have liked to have seen explanation of how microprocessors interface with equipment and each other, along with a discussion on software development.

P. A. BENNETT
Brigg

Continued on p. 494