*Information Retrieval*, edited J.-L. Vidick, pp. 25–44. ACM, New York (1990).
48. F. Rabitti and P. Savino, Image query processing based on multi-level signatures. In: *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, edited A. Bookstein, Y. Chairamella, G. Salton and V. Raghavan, pp. 305–314. ACM, New York (1991).
49. IEEE. *IEEE Data Engineering* **12** (2). Special Issue on Imprecision in Databases (1989).
50. A. Motro, Accommodating imprecision in database systems: issues and solutions. *IEEE Data Engineering Bulletin* **13** (4), 29–42 (1990).
51. N. Fuhr, A probabilistic framework for vague queries and imprecise information in databases. In *Proceedings of the 16th International Conference on Very Large Databases*, edited D. McLeod, R. Sacks-Davis and H. Schek, pp. 696–707. Morgan Kaufman, Los Altos, CA (1990).
52. F. Rabitti and P. Savino, Retrieval of multimedia documents by imprecise query specification. In *Advances in Database Technology – EDBT '90*, edited F. Bancilhon, C. Thanos and D. Tsichritzis, pp. 203–218. Springer, Berlin et al. (1990).
53. A. Bookstein, Information retrieval: a sequential learning process. *Journal of the American Society for Information Science* **34**, 331–342 (1983).
54. W. B. Croft and R. H. Thompson, I3R: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science* **38** (6), 389–404 (1987).



# Book Review

NELL DALE and CHIP WEEMS
*Introduction to Pascal and Structured Design*, 3rd edition
D. C. Heath and Co., Lexington, 1991
925 pp. 0-669-20238-X.
NELL DALE and SUSAN C. LILLY
*Pascal plus Data Structures, Algorithms and Advanced Programming*, 3rd edition
D. C. Heath and Co., Lexington, 1991
850 pp. 0-669-24830-4.

Any book that comes out in a third edition in the cut-throat world of first-year Pascal texts deserves close scrutiny. These two books are a sample from a suite of six books, two laboratory courses and a video course that Nell Dale and colleagues have published through D. C. Heath since 1983. They have evidently refined their winning formula to the point where it is a 'success at over 1250 schools' (quoted from the publisher's advertising). So what is it that the customers like? Why are these books so successful?

There seem to be two factors involved: pedagogy and quality. Both books cover their topics absolutely thoroughly, explaining each new feature as it arises, using interesting examples, both classic (binary search) and novel (absenteeism pattern). Some of the examples progress through the book, becoming more sophisticated as the student's expertise increases, and there are several larger case studies. The order of topics has been carefully chosen, and the authors have adapted the material through each revision so as to reflect changing trends. The teaching and learning aids are impressive: the introductory book concludes each chapter with case studies, advice on testing and debugging, and then four levels of exercises for the students, from a quick check quiz to genuine programming problems. Beginners will find these very reassuring. The advanced book ends each chapter with an extensive set of exercises, and has a separate section at the end of the book with nearly 30 programming assignments. These have obviously been class tested and the attention to detail makes it possible for the lecturer to set them for a class as-is.

Through the editions, the authors have had the time to add in additional material which may be regarded as peripheral, but which, in my opinion, greatly enhances the value of these books. The introductory text is particularly good in this regard, and includes pen sketches of famous computer scientists, snippets of related theory, advice on style and guidelines for program design. The programs are all of a consistently high standard, using plentiful comments, good type definitions, and procedures with parameters to the full. The advanced book includes a diskette containing all the programs, which is a very good sign that the programs have been well tested. Both books are produced in two colours, with cartoons and many graphic illustrations. Quality indeed!

The introductory book has 17 chapters, which take a genuine beginner through the programming process, design methodology and problem solving, to simple Pascal with control structures, procedures and parameters, on to functions, data types and recursion. Purists can note that looping is taught with **while** statements and that **repeat** and **for** only appear later on in the data types section. Subranges are emphasised from chapter 10 onwards, and I was glad to see that they come before arrays. Unlike many books that dive into array handling at the start, Nell and Weems take this conceptually difficult hurdle slowly. Chapters 11, 12 and 13 introduce array processing, patterns of array access, lists, strings tables. Chapter 15 covers files and pointers and chapter 16 provides a gentle introduction to the advanced book on data structures.

The only problem with this book is that it covers standard Pascal and only standard Pascal. Thus many common operations (reading in a string) are long-winded, some (e.g. assigning a file name) have to be side-stepped, and the important programming issues of screen and graphics handling, separate compilation and objects, are simply ignored. There is a companion volume which is a Turbo Pascal version, and one hopes that these aspects are fully addressed there.

Because the advanced book follows the same easily accessible style and high-quality layout of the introductory text (complete with jokes), I was tempted to regard it as not a serious contender in the Data Structures and Algorithms text stakes. Closer scrutiny proved me wrong. Dale and Lilly have managed to retain a high level of rigour in the presentation of abstract data types while still making each of them truly usable in practice. Stacks, queues, lists and binary trees are covered in detail, and the efficiency of each of the algorithms applied to them is formally discussed. Sorting and searching (including hashing) receive similar treatment. However, the book stops short of the next level of ADTs – bags, sets, B-Trees, directed graphs. Turbo Pascal is made use of in places where the authors were really desperate, and there is also a Turbo Pascal version of the book (but how 'Turbo' it is, I do not know).

So, how would these books fit into first courses outside the US? The introductory book is too basic and too slow for a typical first-semester course, where the majority of students will have programmed at school and may even know Pascal. The insight into programming is fine, but is dispersed throughout the text, and overall the book is just too big for the subject it covers. On the other hand, the pedagogical aids are first class, and I would certainly advise any student who is having difficulty with a Pascal course to get this book as back-up. The advanced book is also hefty, and although thorough, does not go as far into the exciting aspects of algorithms as the more formal and classic texts do. On the other hand, the examples and assignments are first rate, and I am very glad to have this book on my shelf.

These are both excellent books in all respects except coverage, and I would certainly recommend that any teachers of first or second year should inspect copies and judge whether the coverage is sufficient for the courses under consideration. If it is, then both class and lecturer will be joining the other 1250 satisfied schools, and are on to a winner.

JUDY BISHOP
*Pretoria*