

The final section of the book claims to "contain material for people with specialized needs or interests (page 281)." It does give some interesting suggestions on how to trace artwork, as well as advice on how to modify the Macintosh™ for use in a moving vehicle (bolt it down.) It also gives simple advice for use by the handicapped. There is a detailed discussion of communications and data-moving concepts, all of which are a little out of place because they are so much in the abstract. The Technical Topics chapter is the gem of the entire book because it contains exceedingly arcane information such as pinouts for the mouse connector and the wiring for an adapter cable to RS-232C devices. These alone could justify the purchase of the book for the really experienced MacUser.

All in all, this is a professionally produced work. It suffers a bit from trying to please two vastly different audiences (those who have seen the MacWay, and those who haven't ...yet), but it remains the most thorough and objective source on information yet available.

\*St. Peter's College, Jersey City, NJ 07306

\*\*Texas A&M University, College Station, TX 77843-2468

**VISICALC FOR SCIENCE AND ENGINEERING**  
 by Stanley R. Trost and Charles Pomernacki  
 SYBEX, Berkeley, CA  
 1983, paperbound, \$13.95

**Reviewed by Harry E. Pence  
 and Celene DiFrancia\***

Although VisiCalc is best known as a spreadsheet program for business applications, it can also be extremely useful in some chemical problems. Especially in analytical chemistry laboratories, it is used to do repetitive calculations, organize and correlate data, develop reports, and identify data trends. In fact, one of the reviewers (C.D.) has done exactly this type of work in an industrial laboratory during the summer and vacations for the past year.

Trost and Pomernacki have written this book to demonstrate some of the possible scientific uses of VisiCalc. The authors presume that the reader already has a working knowledge of VisiCalc, although a brief summary of the necessary commands is provided in an appendix. This assumption should create few difficulties since there are several books available that describe VisiCalc and other similar spreadsheet software.

The major emphasis of the book is on creating design tables or spreadsheets that can be manipulated with routine VisiCalc commands. The sample spreadsheets discussed are intended to both serve as a library of useful applications as well as demonstrate how the techniques can be extended to other situations.

The authors have chosen to draw their examples from a widely divergent group of scientific fields, including not only statistics, communications, heat flow, optics, and simple mechanics, but also electrical, solar, mechanical, and civil engineering. Over fifty examples are provided, and in each case the approach is the same. First a specific situation is outlined, then the appropriate data is provided, and finally instructions are given for a sample problem. The instructions are clear, although a little dry. No unexpected results were discovered in running a number of these examples.

It is unfortunate that so few of the problems selected are directly related to chemical applications. More important, the focus on specific situations tends to somewhat blur the generalities that might identify broader implications of the method. Reading this book will help to develop a general appreciation of the power of VisiCalc and similar spreadsheet analysis software, but it is left to the individual to determine how these principles might relate to chemical problems.

The chemist who wishes to pursue the possibilities of spreadsheet analysis further is left with few alternatives. A brief bibliography is included, but few of the books listed seem likely to deal with chemical problems. In short, the reader must accept the fact that further work must be done independently.

Obviously, this book leaves something to be desired for chemists, but as the old saying goes, it may be the only game in town. Despite the lack of chemical applications, the book can suggest possible chemical uses of VisiCalc. For some readers, the book may be a worth-while source of information and ideas, but others may choose to work on their own or wait until a more directly applicable book becomes available.

\*Department of Chemistry  
SUNY-Oneonta, Oneonta, NY 13820

## **MODULA-2 FOR PASCAL PROGRAMMERS**

by Richard Gleaves

Springer-Verlag, New York, NY  
1984, 145 pages, softcover, \$16.95

**Reviewed by Brian Pankuch\***

Modula-2 is Niklaus Wirth's newest programming language. Pascal, his first language written in the late sixties, was designed principally for teaching structured programming. Somewhat to Wirth's surprise, people began using Pascal as the language of choice in many programming applications. As might be expected, Pascal does not contain many of the ingredients needed to make it a general purpose language. Modula-2 is a general purpose language designed primarily for writing software systems. It introduces changes which allow simpler programming with improved readability and efficiency. The scope of the language is greater than Pascal, since it has been extended to include system design and machine-level programming.

As indicated by the title, this book assumes a knowledge of Pascal. It is divided into three main parts. The first, "New Concepts", was the hardest for me to understand since, of course, it was the newest. All parts of the book are written with many short programming examples, but in this section they were not sufficient for me to develop a good feeling for why each one of the innovations was being made. If I had Modular-2 available, I'd play with the short programs and make a few changes in them to see what happens. This should make the concepts more concrete and understandable.

Modules are similar to units in Pascal. The objects imported and exported to modules must be more carefully defined and are made visible only where needed. This limits the scope of imported objects and makes a module or program that uses the module less likely to give unexpected results.

Modules also seem as if they might be easier to maintain, since it is possible to change some modules without recompiling others in the system. In Pascal, changes in a unit that is called by other units requires each of these units to be recompiled and replaced in the library. It is less than clear to me exactly when this is necessary in Modula-2, but it does appear that the library would be recompiled and restructured less often.

Easier access to machine level programming is important in speeding up a program. Finding where the program is bogged down and optimizing that portion of the program in machine language generally produces significant speed-ups in program execution. This is especially noticeable in graphics and applications involving extensive calculations.

The second part of the book, "Differences from Pascal", was much easier for me to follow. Virtually all of the changes made sense to me. Most of these improved program clarity or ease of use. There were several specific examples that struck me as I was reading. Modula-2 is case sensitive, and so using upper or lower case makes a difference. For instance in Pascal, `FIRSTNAME`, `firstname`, or `FirstName` would all be the same variable; in Modula-2 each is different. The last version, `FirstName`, is the preferred form. All characters in a name are significant, not just the first eight as in most versions of Pascal. All reserved words (those defined in Modula-2) must be capitalized. The result is that many techniques Pascal programmers have normally used to make their programs more readable and legible are now required in order for the program to compile.

`IF`, `FOR`, and `WHILE` have their own `END` instead of being used with `BEGIN-END` pairs. Comments can be nested without having the compiler write strange error messages or commenting out critical parts of a program. Strings and characters can be used more interchangeably. String utilities are provided that include all those used in UCSD Pascal, so the result is not only more flexibility in using strings, but also more freedom.