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. Somehwat 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 is UCSD Pascal, so the result is not only more flexibility in using strings, but also more freedom.

A new type, CARDINAL, has been added. It is an unsigned integer and is used in place of the INTEGER, which is now used only when a VARIABLE can be negative. This would appear to allow the compiler to do more checking if the programmer will initially think out the variable possibilities.

The CASE and variant RECORDS now include an ELSE statement, which will catch unspecified case values. This is handy when using a case selector which is of an unallowed type but isn't specified in the CASE label list. Other changes, such as putting vertical bars in front of CASE label, make the program more readable.

The different types CARDINAL, REAL, and INTEGER are not permitted in the same expression, so it is necessary to decide on the type of the answer and then change all types to explicitly agree with this one type. Fortunately, utilities are included which allow one to change explicitly from one type to another.

There are several other helpful changes. ELSIF allows modifications to the IF THEN statements, so that the FOR statement can have a step value other than 1 (which is still the default value). Procedures can be called before they are declared. File handling seems much more sensible and more specific, with more checking and probably less possibility for error.

The last section contains the utilities. These seem to give much more control over handling errors as well as string handling that is, as noted earlier, very similar to UCSD capabilities. All the input-output procedures for text file or terminal access seem well thought out with more error checking than I'm used to, although in actual use they may not perform as well as expected.

Overall, the book offers many hints and warnings about how to use or not misuse the various aspects of Modula-2. Those who have had previous experience with Pascal should probably be writing simple programs in a few hours with this book. I would recommend beginning by going through the first and third parts very rapidly to gain a general overview. After that, it would be best to concentrate on part two, actually write some simple things, and play with some of the examples. Learning the syntax should occur quickly since it is very similar to Pascal. For the newer concepts I think it would take me a lot more than a few hours and probably more than what is covered in this book to understand them thoroughly.

For those who are thinking of getting involved with Modula-2, I'd recommend reading Gleaves' book before investing in a Modula-2 compiler. The book explains better than anything else I've seen what Modula-2 is all about. Although it takes some work, it is certainly a readable text.

*Chemistry Department
Union County College
Cranford, NJ 07016

THREE DEGREES ABOVE ZERO: BELL LABS IN THE INFORMATION AGE

by Jeremy Bernstein
Charles Scribner's Sons, New York, NY
1984, 241 pages, hardcover, $17.95

Reviewed by Brian Pankuch*

This book is an intriguing mixture of the history of technology and science with personal insights about some of the scientists who have made important discoveries. Bernstein provides extensive background material about many Bell scientists, beginning with their early research interests and continuing to the current work they are doing at Bell Labs. I was struck by the diversity of these individual stories. While some scientists seemed to move directly into their main research, others knew what they wanted to do but found it difficult to pursue their chosen field at the appropriate level until arriving at Bell. A great deal is said about academic freedom in higher education, but one can't help but be impressed by the freedom many of these scientists feel they have. The combination of basic research, technology, development and the freedom to pursue personal interests is not only satisfying to the individual but also a very successful approach. Seven Nobel Prizes in Physics alone have been won by Bell Lab scientists.

The four main parts of the book are mostly concerned with the solid state and fiber optic areas, but considerable chemistry and material science seep in. Many sections of the book offer impressive descriptions of the latest technology. Bell is testing fiber-optics which transmit 420 million bits/sec. Using these techniques, a thirty volume encylcopedia could be transmitted in one second, making a mistake in only a single letter.