I have no listing of Tiny Basic and the book I have from Netronics is very vague and undetailed. I talked to Netronics but no help.

Tiny Basic, or maybe how I could get a program listing with good comments.

My Basic questions are:

- **Routine**
- **Read to**
- **_I_** (con-)
- **_y_**
- **So the**

Kilby

g cor-

is that

the

LEN

block

other

he gi

---

mon

ger-

code.

T

do-

but

McC-

McCr

TRA

is r

com

exp

tec

new

R

Rumage through your parts junk box and locate an old filter choke: an old T-V choke is ideal. Since the majority of these have the E and I laminations in separate stacks, it's easy to remove the core mounting frame and pry the I laminations loose. This leaves you with the E laminations and the coil in the center. Mount the E laminations/coil in a small plastic or aluminum box by modifying the old mounting frame or using epoxy glue. Add a switch and a line cord and you're in business. A high wattage resistor (about 10 watts) may be needed with some low resistance coils to prevent excessive current draw. Try to pick a choke with a D-C resistance of approximately 30 ohms. This should provide a field strong enough to thoroughly obliterate all programs.

Turn the eraser on and slide the cassette over the open lamination section—a slight circular motion is desirable. After 10/15 seconds, remove the cassette and turn off the eraser. Operation for more than a minute tends to heat the coil up a little, so avoid continuous operation. By the way, the discarded I laminations are useful in concentrating the eraser's magnetic field to demagnetize tape heads.

Sincerely,

Controlex Engineering

Tom Hamilton

Box 473

Birmingham, MI 48012

---

**Book Reviews**

**Increasing Programmer Productivity Through Logic Development**

Gloria Harrington Swann

PBI-Petrocelli (McGraw-Hill)

$12.50

If this were one chapter of a larger book, I would praise it strongly as an excellent treatment of what it covers. Standing alone, one wonders if it's worth the money. The book is only 116 pages, including the index and a short glossary; the pages are rather small and over half of them are primarily flow charts or similar diagrams.

Some introductory texts include a discussion of typical business data processing applications using sequential files. This book is almost entirely devoted to discussing and flow-charting these traditional functions—e.g., updating a master file from a transaction file, printing subtotals when there is a change in one of the control fields. The meat of the book is in carefully drawn and explained flow charts and Chapin charts (a helpful and well-described variant of flow charting) for seven basic file-processing functions; introductory sections discuss how to construct flow charts, Chapin charts, and decision logic tables.

For the beginning programmer starting to construct simple systems for himself, or coming in to maintain existing systems, the examples and explanations should be very helpful. They are an excellent library of clearly written standard routines for standard functions, with some examples given of how to pace them together for more complex functions.

The language is readable and informal (enough to grate on the trained ear at times: is the "do as long as ..." construction on page 18 really more useful to the novice than "do while ..."?), specific computer languages are used in the text ("COBOL" is an entry in the glossary but not the index) but COBOL is clearly the language in mind (a flowchart entry on page 76: "Move all 9's to project number in master").

Who, then, is this book for? If you need an elementary introduction to traditional applications of sequential files, or if you once heard 3 lectures on flow charts but are not yet convinced that they convey useful information, the book will certainly help. It is probably very appropriate for high school libraries. But it does not, in the end, satisfy the claims implied by the title and the dust jacket description. For instance, the chapter "Common Programming Errors" is only two brief pages; "Coding Standards" is three. One hopes that the reader will not be misled into thinking such things are so simple. For instance, Dennis Van Tassel's excellent Program Style, Design, Efficiency, Debugging, and Testing (Prentice-Hall, 1974) devotes 32 pages to style and 49 to debugging; but that book lacks the flow charts and business motivation, and might be too advanced for the beginning programmer who is just ready to benefit from Swann's book.

Reviewed by

Edward T. Ordman,

New England College.

**Learning Level II**

By David A. Lien

Publisher: Compusoft

"352 pages"