BOOK REVIEWS

Programming microcomputers for psychology experiments

R. DENI
Belmont, CA: Wadsworth, 1986
262 pages (paperback)

The following review is by N. John Castellan, Jr., *Indiana University, Bloomington, Indiana*

Programming Microcomputers for Psychology Experiments is designed for use in undergraduate psychology courses that teach computer programming for psychology experiments that may be run on microcomputers. Although there are books on real-time computing available for psychologists in which the uses of computers in experimental research are discussed, to my knowledge, none has been devoted solely to microcomputers. Despite the fact that Deni's book is designed to fill a gap, this reviewer has some concerns about the appropriateness of the book for classroom purposes.

The book teaches programming by having the student design an increasingly elaborate paired-associate experiment. This incremental approach to teaching programming has been used successfully in other behavioral science programming texts (e.g., Lehman, 1977). The experiment begins with a very simple program that evolves into a complex program with response time collection and data analysis. This pedagogical tool is excellent, and I wish that the problems with the book had been minor enough to not compromise this approach.

The book is divided into four parts and contains 13 chapters and appendices. Part 1, "Getting Started," consists of three chapters: (1) "Microcomputers in the Experimental Psychology Laboratory," (2) "Fundamentals of Disk Operating Systems," and (3) "Fundamentals of Programming in BASIC—BASIC Commands." These chapters provide an elementary introduction to the subject. Part 2, "Applications," consists of four chapters: (4) "Fundamentals of Programming in BASIC—Statements," (5) "Fundamentals of Real-Time Computing," (6) "Measurement of Subject Behavior," and (7) "Control of Stimulus Presentation." These chapters begin with elementary BASIC statements and build on them in the context of the paired-associate learning experiment to teach the code necessary to program any experimental procedure. Part 3, "Refinements," has three chapters: (8) "On-Line Storage and Display of Subject Data," (9) "Off-Line Disk Storage of Subject Data," and (10) "Programming the Experimental Design." Part 4,

"Advanced Programming Projects," contains three chapters: (11) "Special Projects in Verbal Learning and Memory," (12) "Low-Resolution Graphics and a Special Project in Cognition," (13) "High-Resolution Graphics and a Special Project in Perception." These chapters proceed through several useful examples of programming techniques for students. The projects are reasonable and suitable for an introductory laboratory methods course. The programs would be less successful in advanced courses; one reason is that generalization is made difficult because the author has failed to take a structured programming approach in program design.

To address the book to as broad an audience as possible, the author discusses programming for the Apple II, the IBM-PC, and the TRS-80. All of these machines are used in laboratory work; however, as the author correctly points out, there are significant differences among the operating systems and BASIC languages on these machines. These incompatibilities pose a major challenge to the author in his attempt to satisfy three different audiences. His solution is to discuss all three machines in a combined serial and parallel presentation. As concepts are introduced, differences between machines are discussed serially within paragraphs and in parallel within tables. The text presentation suffers because the same (or similar) functions on the three machines are described within a single paragraph, sometimes even within a single sentence. The tables are useful but awkward to follow because the machine differences are sometimes arranged by columns and sometimes by rows.

This mixed mode of presentation makes the text extremely difficult for a student who is trying to learn to program on only one machine. However, for the researcher who is translating an existing program from one machine to another, a convenient table and compilation of differences between operating systems and BASICs can be quite valuable indeed. Unfortunately, the introductory level at which the book is written makes such use of the text by the advanced researcher rather limited.

A serious problem with the text is that *no* books or papers are referenced. Thus the reader has no idea where to go for the further study that he or she will surely need. There is extensive literature on using computers in psychological research and experimentation. Some books date back nearly 25 years but still contain relevant information (Apter & Westby, 1977; Cooper, 1977; Green, 1963; Lehman, 1977; Mayzner & Dolan, 1978; Uttal, 1968; Weiss, 1973). There are also many technical books that are invaluable to the researcher programmer (e.g., Luebert, 1982). Surely the author could have cited some books and papers. Although the author encourages his reader

to read *Behavior Research Methods, Instruments, & Computers*, I am convinced that had the author taken his own advice, the book would have been far better.

Computer Structure. One is surprised to find no functional description of computer memory and disk storage. Descriptions often are given by analogy. This is satisfactory when the analogy is useful; however, the analogies can lead to confusion when one attempts to apply what has been learned. An example is evident in the following quote: "If you are confused about where your program actually *is* in the computer at any point, let me summarize the situation" (emphasis added). However, the author never describes the situation, and the summary that follows never really says "where" the information is. There is no distinction between when a program is in memory and when the program is on disk and no discussion about the relationship between what is on disk, what is in memory, and what is on screen. These distinctions are critical to people who are trying to understand computers.

Another technical problem is that the author initially describes a file as a *unit* of information. Later, he states that a file may contain a program or sets of text or data. While perhaps technically correct, the description may be somewhat confusing to the beginning student.

Program Design. Another problem with the text is that the programming techniques that are taught differ significantly from current programming practice. There are no flowcharts in the text, and no equivalent technique is proposed. In addition, although the incremental approach to programming used by the author may be useful, the text makes no use of structured program design. As a result, some of the coding is cumbersome and difficult to follow. An example of programming that is difficult for students to follow and that is poor programming style is the following code for counting leverpresses:

```
300 LEVER = INP(SLVB)
310 IF LEVER = 0 THEN 300
320 COUNT = COUNT + 1
330 GO TO 300
```

A clearer representation would be the following:

```
300 LEVER = INP(SLVB)
310 IF LEVER > 0 THEN COUNT = COUNT + 1
320 GO TO 300
```

Or, in order to avoid (eventually) the endless loop in the original coding by allowing multiple branches,

```
300 LEVER = INP(SLVB)
310 IF LEVER = 0 THEN GO TO 300
    ELSE COUNT = COUNT + 1 : GO TO 300
```

These recodings, although still containing undesirable GO TO statements, are clearer, even when REMARKS are added to the code.

Hardware Dependence. One difficulty found in reviewing these chapters is that in the measurement of response times, the author assumes that the user has a particular interface (distributed by LVB) that will provide

the necessary timing. This interface is one that must be purchased and that differs from other commercial interfaces. Moreover, much of the discussion of the interface is a summary of what is contained in the interface user's manual. Consequently, much of the discussion on timing is limited, since the emphasis is put on a particular external device. That it is possible to do timing using clocks within the machines is not pointed out clearly. Routines to do this are readily available and have been described in *Behavior Research Methods, Instruments, & Computers*.

Program Accuracy. One of the more serious problems with the text is the number of coding errors that could result in flawed experiments. One example is the generation of random numbers. Stimuli (and other data) that have been randomized are used extensively in experimental psychology, and it is essential that researchers have good randomizing procedures. The author discusses permuting a list of elements and describes an algorithm that samples all elements with replacement from an array and checks to determine whether a sampled element has already been sampled. If the element has been sampled, another sample is taken. A dummy array is used to keep track of which elements have been sampled. First, this is an extremely inefficient procedure. For example, when permuting an array of 10 elements, when one gets to the last 2 elements, one is resampling previously sampled elements 80%-90% of the time. The execution time of such generators increases *geometrically* with the size of the array. Second, and perhaps more importantly, it is known that this procedure does not ensure a random permutation. Correct techniques for which execution time varies only linearly with the number of elements in the array are found in many behavioral science texts on computing (e.g., Lehman, 1977).

Omissions. There are some important topics missing from the book. In experimental control and in data analysis, it is frequently necessary to sort arrays. This topic is not covered at all. Also, the author tries very hard to avoid teaching students about the binary coding of numeric quantities. For example, in discussing the PEEK and INP functions for testing channels in the interface, a discussion of binary coding would have made the process much easier to understand. One might grant that if binary coding were relevant only for that particular function, it might well be omitted; however, in discussion of other inputs and other functions, the author continues to talk around binary coding of variables, which, in the long term, may confuse rather than help students. The binary coding situation is partially resolved in a table on page 89. The author indicates near the table that he will refresh the reader's memory; however, the memory cannot be refreshed because, based on the text, there is none to be refreshed. Moreover, the author does not explain that binary coding is a useful concept that is used (explicitly or implicitly) elsewhere.

Summary. I used this book with a group of students who were learning real-time computing and found that they had a very difficult time with the book. It is not well

suited for self-study, and an instructor who uses it should be prepared to spend a great deal of time clarifying material in the text. It might be used as an adjunct to a course, but the instructor needs to be very careful to tutor good programming skills and to counter many of those presented in this text. The increasingly sophisticated students, faculty, and researchers in experimental psychology deserve texts and handbooks to meet their needs. The teacher wishing to teach real-time computer-based experimental programming design must continue to use a variety of judiciously chosen references until a comprehensive and accurate text is available.

REFERENCES

- APTER, M. J., & WESTBY, G. (1977). *The computer in psychology*. New York: Wiley.
- COOPER, J. W. (1977). *The minicomputer in the laboratory*. New York: Wiley.
- GREEN, B. F., JR. (1963). *Digital computers in research: An introduction for behavioral and social scientists*. New York: McGraw-Hill.
- LEHMAN, R. S. (1977). *Computer simulation and modeling*. Hillsdale, NJ: Erlbaum.
- LUEBBERT, W. E. (1982). *What's where in the APPLE: A complete guide to the Apple computer*. Chelmsford, MA: Micro Ink.
- MAYZNER, M. S., & DOLAN, T. R., (Eds.). (1978). *Minicomputers in sensory and information-processing research*. Hillsdale, NJ: Erlbaum.
- WEISS, B., (Ed.). (1973). *Digital computers in the behavioral laboratory*. New York: Appleton-Century-Crofts.

The following review is by David A. Eckerman, Angela Allen, Pamela Doty, Sheryl Scharf Moy, and Beth A. Zech, *University of North Carolina at Chapel Hill, Chapel Hill, North Carolina*

Prologue (by D.A.E.). Early during the fall semester, I sat down with a group of first-year graduate students in the experimental and biological psychology program to plan how they might learn to use computers in their work. I would bet that several similar groups started out this fall in other schools. Like us, they wanted to encourage training without taking too much time from an already full schedule. Like me, each leader hoped it could be accomplished with a minimum of his or her time. The new book by Deni seemed ideal. It is brief and offers all the apparent advantages listed below. We saddled up with topics for a dozen weekly meetings, made sure we packed a microcomputer, and settled into what seemed a pleasant journey. Well, we never made good progress on that trip. A new start was needed during the spring semester. The Deni book did not provide the support we had anticipated.

In his book, Deni tries to accomplish the following: assume no special preparation in computers; allow use of an Apple II, Tandy TRS-80, or IBM-PC computer; teach BASIC programming; teach about operating systems as well as a programming language; integrate learning laboratory interfacing with "stand alone" uses; allow

interfacing projects, however, to be "passed over"; use psychology lab examples; build good habits of data backup; pretest material before publication; and write in a light, easy-going style.

Deni is successful in providing good example programs, offering a useful guide for instructors, emphasizing use of subroutines, recommending helpful programming safeguards, and keeping the reader focused on psychology applications.

The shortcomings of Deni's book are that it offers no "road map" for concepts covered in each section; defines in a "hit and run" fashion; does not tie in well with reference material; does not distinguish important information from detail; and focuses too much on verbal learning and memory.

Review. Our criterion in judging Deni's book was as follows: Could a relatively unled group of psychology graduate students who were fairly naive about computers use this book, supplemented with consultation requested on the basis of individual readings, to incorporate computers into the control of their experiments? After a semester of trying this strategy, we decided that they could not. On the other hand, the book might very well be useful to an instructor of such a group and/or to those with a background in BASIC programming.

A review of the book's chapters might be a useful way to describe the book's strengths and shortcomings. Chapters 1-3 offer an overview at a very elementary level. Deni appropriately assumes that many students will have little or no prior computer experience and makes his introduction refreshingly easy. Remarkably, we discover in Chapter 1 that Deni later introduces us to *three separate computer systems*. For a slim volume, this seems too optimistic. Yet the goal is not as unrealistic as one might at first think, since Applesoft BASIC, TRS-80 BASIC, and IBM-PC BASIC are all scions from the same Microsoft BASIC tree. Their operating systems are not as closely related, but Deni does not presume to teach very much about resource or file handling. Thus, Chapter 1 leaves even the experienced reader with confidence that the goals are achievable for readers at all levels. Chapter 2 provides a very brief introduction to operating system activities, such as accessing disk drives, starting up, making backups, and getting directories. Although his descriptions are brief, the examples are comprehensible for they have limited, practical goals. Chapter 3 starts off teaching the reader some "housekeeping" BASIC commands: NEW, LIST, RUN, and SAVE. Descriptions are again brief, but the reader can follow them since the goals are limited and practical. There are, however, a few issues one might stumble over. Deni does not use a clear method for differentiating between what is typed in by the programmer and what is a computer response. The use of parentheses, dot series, and other notation is somewhat confusing. Deni indicates that reserved words cannot be

used as file names but does not specify what these words are or how the reader might find such a list (more on similar omissions later).

With Chapter 4, our mood changed. We were now ready to learn BASIC programming. Deni offers no clear statement of purpose or approach. Are we to learn BASIC programming here? If so, we will need a clear development of the statements and functions in an order that allows us to incorporate them into a general approach to programming. Are we to merely see examples of BASIC programming applied to a few psychology examples (which turns out to be the case)? Then the author should provide direct ties to reference materials that will help us learn programming. Either approach would be of use, but Deni does not offer a "road map." He begins by telling about constants and the DATA statement. Why such a brief description of variables? Why are we not given a chance to see where these statements will ultimately fit into our work? How are the sections sequenced? At last we conclude that Deni is only teaching us how to display 10 words. He then touches on READ, LET, INPUT, PRINT, FOR-NEXT, print formatting options, and REM. The order is seemingly haphazard.

The level of Deni's description is also a problem. The Chapter 4 descriptions remain brief and are as example-specific as they were in Chapters 2 and 3. Sufficient detail is not provided for the reader to visualize commands and functions in their general use. For example, LET and assignment is downplayed as though it were used mostly to perform arithmetic. FOR-NEXT is described in 240 fairly casual words (e.g., "[These terms]... perform what are called loops in computer lingo" [p. 42]). Powerful general computer statements seem only as important as CLS, HOME, and VTAB. Furthermore, the subject matter is never tied to reference material provided for individual machines. The reader should be encouraged to locate complete descriptions, but he/she is not. As a case in point, the danger of *reserved words* is briefly noted, but the reader is given no help in finding a list of reserved words. Instead, a caution is given that "You have to learn to watch out for these little pitfalls" (p. 37). One important omission concerns editing commands. Deni's only discussion of editing is an explanation of DEL, which is followed by an admission that it is easier to merely type in a line number alone. No encouragement is offered for the student to find out about line editing aids.

By Chapters 5 and 6, Deni hopes that the readers have conquered enough BASIC to allow them to handle input from an external timer and switches. The two introductory pages of Chapter 5 seem just right, and at first, we were persuaded by Deni's statement in the "Preface" that the readers could treat the interface as a "black box" and proceed, even if they did not have access to this device. We now feel different. The treatment of I/O from the interface is very different from that required for formatting screen displays, but Deni does not prepare us well for these differences. For computer input and output, Deni relies on the Med Associate's LVB interface, "because

it can be connected to each of the three computer systems." Although it is a perfectly adequate interface, we wonder why this particular (expensive) device is used. Be advised that the unit is treated neither as a "black box" nor as a "generic interface"; instead, Deni dwells on particulars. He encourages the readers to assemble their own LVB interfaces before proceeding. This suggestion is now famous in our group, for his description of how to approach this task is understandable only if one has considerable experience fabricating electronic circuits. Deni's ambivalence on this topic is clear when he advises "if you don't feel confident messing around in your computer, call your local computer dealer" (p. 216). Many notions, such as ports, control lines, data lines, memory-mapped I/O, INP/OUT, and PEEK/POKE, are introduced too briefly.

In addition to reading time from the LVB interface, Deni also considers reading time from the computer itself (TRS-80 and IBM-PC) or from a clock card (Apple II). The explanation of the programming examples in this section starts off well. Deni reminds the reader of concepts that have been introduced in past chapters: the FOR-NEXT loop, DATA statement, and centering of a display. The notion of a "latency timer or process timer" is introduced and compared to the "interval timer," which was quickly introduced in Chapter 5, as though to an old friend. The topic is too quickly introduced, however, to suppose that the various timer options are well understood, and we found that the students among us did not fully comprehend their differences. Further explanation of their function would be appropriate. Latency and duration are quickly defined, without real depth or generality.

Chapters 5 and 6 also include an approach to the input of information into a program from the LVB switches and from the computer keyboard. The treatment of switch inputs from the LVB interface is left quite opaque. To obtain switch information, the programmer needs to subtract byte decimal equivalents. The treatment of bit numbers in a byte, however, is left remarkably vague and threatening (pp. 70-71). Deni calls these numbers merely LBV interface "channel labels" without further explanation. Information on keyboard entry is more complete and easier to follow, although differences between the Apple and other systems are handled somewhat awkwardly. Although Deni includes a nice summary of ASCII codes, the description is in an awkward place. An earlier introduction of this important notion would be less distracting.

Chapters 5 and 6 also include descriptions of other BASIC statements and functions (MID\$, VAL, IF-THEN, INKEY\$). For these, Deni continues to give few clues as to the general uses. He does not provide details about what "INKEY\$ assigns the null string" means or what INKEY\$ stands for. (The mnemonic nature of command names is never commented on.) The very important notion of subroutines is also presented too quickly. ("Think of a subroutine as simply a part of a program" [p. 55]).

We should also stress that there is some very good advice in these chapters. For example, Deni warns the reader about clearing unwanted inputs before starting an input sequence. He emphasizes how important it is to print the raw data.

In Chapter 7, matters turn from input to output. He teaches students to center stimulus words on the computer screen, then to generate tones, and finally to activate drivers (or relays) on the LVB interface. The chapter begins, however, with no real overview as to what will be accomplished. While Deni is describing how to center words on the screen, he explains BASIC statements LEN and INT and such activities as string concatenation adequately for the example at hand, but their general usefulness is not fully demonstrated.

Bell and sound functions in the three computers are made fairly clear, and Deni demonstrates why one might want to use them. For the Apple II, it would be useful to reference an article that would fully describe how to generate tones of various frequencies and durations (e.g., Bender, 1982), since that information is not available in material provided by Apple, Inc.

Throughout the book, Deni writes out whole programs in successive "cuttings" of listings. This approach, though understandable, is a bit confusing, especially when bridging across partial listings for the three different computers. When a whole program is actually written in a single listing (e.g., p. 83), it is quite refreshing and provides a quick review, and the reader is helped when actually working on the computer.

The section on LVB output provides appropriate and useful examples of what can be controlled through "pulse" and "level" modes with the LVB device. It is not hard for the reader to extrapolate these output concepts to other types of interfaces. The program example for schedules of reinforcement is a start, but why does Deni not suggest which sections of the program one would change to arrange more complex schedules? The reader is left with a program for fixed ratio 2 (although FR size is an easily manipulated parameter). The example of cue light control enables the reader to understand how different stimuli can be incorporated into the program. Again, however, the reader must deal with LVB specifics (parameter statements and channel numbers), and general interfacing issues are not well addressed.

Chapters 8 and 9 provide information on the storage of data and advanced printing, including the summarization of data in tables. The data array is introduced, but Deni again defines this concept too vaguely. He says data arrays are "quite similar to a variable... [except that they] can hold a series of values" (p. 105). Notice that an array is not so much "similar to" variables as it is an organized set of variables. The term "series of values" is ambiguous since it could be successive or simultaneous. Deni indicates that array element's are identified by subscripts, but his treatment of the term *subscript* when referring to values in the array elements's parentheses is confusing,

and the identification of the first element with row and the second element with column is not, as implied, a predetermined characteristic. The treatment of the subscript as a constant or a variable is also too indirect. Program examples do not systematically identify arrays used for various functions in the examples.

In the section on the DIM(ension) statement, Deni briefly mentions the allocation of array space beginning with the 0th element. Further elaboration of this often mysterious element would be helpful. Also, DIM is not adequately described. The program example following its introduction has no commentary notes for this statement, even though this is a newly introduced command, and the material under the DIM heading deals more with modifications of the semantic differential program being used as an example than with the use of DIM. The discussion on limits of memory available to arrays is vague.

Deni next addresses printing tables from arrays. Given the restricted definition he has provided for the FOR-NEXT statement, greater detail and more precision are needed for the descriptions of loop functions at this stage. For example, it is misleading when he says that the line NEXT causes the next row to be printed. It actually causes the return to the top of the for-next loop. Exact specification of how loops function in a program are, of course, essential for the book to provide an adequate understanding of programming.

The section on statistical computations is straightforward and well presented, except that the use of the caret symbol (^) is not clearly designated and Deni indicates the square root function as "SQR(?)" rather than indicating that a variable or constant can go in the parentheses. The section on rounding off with PRINT USING provides several good examples and adequate discussion. The formula used on the Apple IIe, however, needs more description.

In Chapter 9, Deni stresses the advantages of using disk storage and encourages disk storage using printout formatted sequential-access files. Why so much effort is devoted to the development of print zones for Apple disk storage is not clear. At this point, such detail seems unnecessary. The specification of I/O channel numbers on the TRS-80 and the IBM-PC was not clear to us. The section on the CLOSE statement and subsequent use of append or extend modes was also not adequately explained.

Deni offers counsel regarding sources of error and warns of possible data losses throughout the book. To this end, a separate section was added to Chapter 9 warning about the reuse of file names, and suggesting the use of systematic file names and the use of string variables rather than string constants when generating a file name from a program. Some attention is also given to how one determines whether adequate disk space is available. For Apple computers, however, the recommended program FILEM is not found on all DOS 3.3 master disks. The FID program, which is more common, can also be uti-

lized to calculate free space. There is one more Apple error: the program segment on obtaining the response measures on an Apple IIe is placed out of order (p. 129).

In Chapter 10, Deni is ready to turn fully back to psychology, using the new programming, interfacing, and file-handling skills the reader has developed. He offers advice on matters of experimental design (internal validity, biasing factors, between-subjects vs. within-subjects designs, counterbalancing). Since this is not a text on design, however, the terms are thrown about fairly glibly, and there is not much depth in their treatment. It would be desirable, however, for Deni to provide references to texts in which the terms would be fully treated. His recommendation of "program templates" is a sound approach for handling independent variables that are changed between subjects or phases. The author also gives an approach to randomizing program conditions for within-subject designs, but there is a thicket of variables, arrays, and machine differences to take into account here. We wish he had used some graphic aids in tracking where the true value of the variable was being kept at various stages in this process. Also, it is unfortunate that he selected a program example that required the LVB interface. The notion of seeding a random generator was left fairly vague.

We followed Deni into Chapter 11, but not beyond, as his remaining two chapters deal with graphic display projects that did not interest our group. Chapter 11 provides one complete example program on verbal serial learning and memory and another on Sternberg memory scanning. These two examples build nicely on each other and are perhaps the best work in the book. They demonstrate how many common tasks the experimental control programs share, and how developing one program aids in developing the next. That our group was hoping Deni would provide comparable examples for other psychological research topics confirms, however, that we were not yet comfortable enough to be ready to tackle those tasks on our own.

REFERENCE

- BENDER, J. H. (1982). Pitch and rhythm on the Apple. *CALL-A.P.P.L.E.*, 5(6), 15-17.

Preparation of this review was supported in part by USPHS Grant MH-18369 to the Experimental and Biological Psychology program.