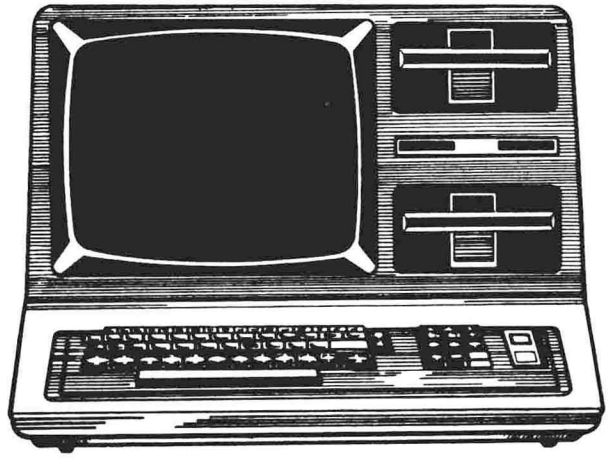
INSTRUCTIONAL COMPUTING

WITH THE

TRS-80®

HERBERT NICKLES GEORGE CULP

INSTRUCTIONAL COMPUTING WITH THE TRS-80®



Herbert L. Nickles
Coordinator of Instructional Computing
Computer Center
California State College at San Bernardino

George H. Culp
Assistant Director for Instructional Computing
Computation Center
University of Texas at Austir

Brooks/Cole Publishing Company
Monterey, California

Brooks/Cole Publishing Company
A Division of Wadsworth, Inc.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging in Publication Data

Nickles, Herbert.
Instructional Computing with the TRS-80.

(Brooks/Cole series in computer education)
Bibliography: p.
Includes index.
1. TRS-80 (Computer)—Programming. 2. Basic
(Computer program language) 3. Computer-assisted
instruction. I. Culp, George H. II. Title.
III. Series.
QA76.8.T18N5 1983 001.64 83-10160
ISBN 0-534-02966-3

Sponsoring Editor: *Michael Needham*
Production Editor: *Richard Mason*
Manuscript Editor: *Dex Ott*
Interior and Cover Design: *Vernon T. Boes*
Cover Photo: *Stan Rice*
Art Coordinator: *Rebecca Tait*
Typesetting: *Graphic Typesetting Service, Los Angeles, California*

PREFACE

"The PTA at our school has given us two microcomputers. Since I am a math/science teacher, the principal said I should use them in my classes. I had one course in Fortran programming about five years ago; I know very little about computers and absolutely nothing about how to use them with instruction. I am aware that they are being used as effective teaching tools. But how? What do I need to know to get started?"

An Inservice Teacher

"As part of an education course I'm taking, I recently sat in on a school board meeting of our district. The agenda included a presentation by a man demonstrating the instructional uses of microcomputers. The man's assistant, who operated the microcomputer, loaded and ran programs and, in general, demonstrated the system, was his six-year-old daughter! It is obvious to me that I'll be facing many young students who are using microcomputers at home and in the school. Even more obviously, I'll need to know something about the different uses of micros: What they can and cannot do, how they are programmed, how these programs are made and tested, and so on. But I don't want to become a 'computer scientist.' I want to know the fundamentals that will let me make practical use of a microcomputer in an instructional setting."

A Preservice Teacher

This book is designed for teachers who find themselves in situations similar to those cited in the two examples above. It is based upon a university course in which hundreds of teachers in grade levels elementary through college have been introduced to the fundamentals of the instructional use of computers and have successfully designed and developed programs for use in their own areas of interest.

This book is not designed to teach general computer literacy. There is little, if any, mention of the history, architecture, or widespread use of computers in society. Nor is this a text designed to train computer programmers. Several language statements common to programming texts are omitted because their application is not typical of the *instructional* use of computers.

This is a practical book for the teacher who needs to know the fundamentals of the BASIC programming language for the TRS-80® Model III and 4 microcomputers and how to apply them to the design and development of instructional computing programs. It has been our experience

that, given these fundamentals, teachers subsequently have the proficiency to expand upon this base and develop efficient programs designed to meet their specific needs.

The book consists of nine chapters and four appendixes and is divided into two parts. In Part One, the first four chapters discuss the BASIC programming language statements and commands common to five areas of instructional computing use: problem solving, drill and practice, tutorial dialog, simulation and gaming, and testing. Chapter 5 summarizes and reviews these statements and their applications. Chapter 6 gives relatively short example and model programs in each of these five areas. Chapter 7 discusses and demonstrates the simple use of graphics as an instructional technique.

In Part Two, Chapters 8 and 9 discuss the specific steps needed first to design and then develop instructional computing programs. The appendixes include instructions for "booting up" the microcomputer, instructions for loading, editing, and saving programs, commands and statements unique to the TRS-80® Model III and 4 microcomputers, answers to questions and problems given in the chapters, and an annotated bibliography of journals and other publications dealing with instructional computing.

As a matter of personal preference, some readers may wish to study Chapters 8 and 9 on design and development prior to the chapters on BASIC. We believe, however, that practical design and development can come only after the working guidelines for the language are established. Thus, BASIC fundamentals are presented before the discussion of design fundamentals.

Twenty-six programs ranging from simple introductory examples to more complex instructional computing application models, plus a "keyword" subroutine and a program "menu" routine are presented in the text. Professors adopting the text may write to the publisher for a free copy of the software diskette, which contains these programs along with solution programs to selected problems in the text. On request, the publisher will make copies of the diskette available to students for $10 each.

The authors are indebted to many people for the development of this book. Only with the critical review of *An Apple for the Teacher* by Sister Mary K. Keller of Clarke College, Professor Edward B. Wright of Western Oregon State College, and Professors Dennis Harper and Jeffrey Marcus of the University of California at Santa Barbara, could this book have been published. Larry Hall, Morgan Watkins, and Carey Van Loon also provided their expert assistance. Supervision of production was ably conducted by Richard Mason. And, most of all, we wish to gratefully acknowledge the contribution of over 1000 students who have provided direct input and response during the developmental stages of this book.

Herbert Nickles
George H. Culp

Brooks/Cole Series in Computer Education

An Apple for the Teacher: Fundamentals of Instructional Computing
George H. Culp and Herbert Nickles

RUN: Computer Education
Dennis O. Harper and James H. Stewart

Brooks/Cole Series of Microcomputer Texts

BASIC: An Introduction to Computer Programming with the Apple
Robert J. Bent and George C. Sethares

Structured COBOL for Microcomputers
Keith Carver

Learning BASIC Programming: A Systematic Approach
Howard Dachslager, Masato Hayashi, and Richard Zucker

Basic Business BASIC: Using Microcomputers
Peter Mears and Louis Raho

Personal Computing: BASIC Programming on the TRS-80
Robert R. Hare, Jr.

CONTENTS

Introduction

This text consists of nine chapters that describe an approach to using a common programming language, BASIC, for the design and development of instructional computing programs for TRS-80 Model III and Model 4 microcomputers. These chapters discuss certain fundamentals of the language and the design and developmental processes that provide a foundation for the production of instructional computing programs.

There are more than one hundred books available that teach BASIC (the Beginners All-purpose Symbolic Instruction Code, developed by John G. Kemeny and Thomas E. Kurtz at Dartmouth College). Although most of these books are very thorough in describing BASIC, they usually emphasize problem-solving applications. Our emphasis, on the other hand, is on instruction in the use of BASIC in the design and development of materials for *instructional computing*, which we now proceed to define.

Simply put, any use of computing techniques within the classroom may be broadly defined as instructional computing (sometimes known as computer-assisted instruction or CAI). Specifically, it includes:

1. *Problem solving*, in which computer programs are written to solve discipline-oriented problems.
2. *Drill and practice* on fundamental concepts using computer programs in a given discipline.
3. *Tutorial dialog*, in which computer programs provide "tutorlike" assistance in pointing out certain types of mistakes, providing review if needed, skipping areas in which proficiency is shown, and so on.
4. *Simulation*, in which computer programs allow manipulation and interpretation of certain elements related to given physical or social phenomena without the constraints of time, space, equipment, and environmental or logistical limits.
5. *Testing*, in which computer programs ask the questions, check the answers, and record the performance.

For our purposes, the term *instructional computing* is used to include all of these applications.

The Use of BASIC An introduction to some of the fundamentals of BASIC is provided in this book. This introduction is not intended to produce highly accomplished and skilled programmers. Rather, it gives only the funda-

mentals needed to write simple programs for instructional computing applications. Model programs are described that illustrate this use.

Although many different programming languages may be used in instructional computing, there are several reasons for using BASIC:

1. It is easy to learn and easy to use.
2. It is a common interactive language (see Section 1.3), available on large computer systems costing millions, medium-sized systems costing hundreds of thousands, minisystems costing tens of thousands, and small systems (commonly called micros or personal computers) costing a few hundred to a few thousand dollars.
3. It may be used in all of the applications of instructional computing described above.
4. It is the introductory computer language used in most secondary and many elementary schools.
5. It is the most common language of microcomputers—an area of computer technology that is making the major impact on education in this decade.

Design Following the introduction to BASIC, a method for designing instructional materials called the *systems approach* is outlined. This approach, in essence, is a logical, step-by-step process for identifying the tasks and activities needed in the production of validated instructional materials.

Development The development of instructional computing programs by the reader is the ultimate goal of this book. Initially, the development phase overlaps the design phase in which paper, pencil, and brain power are the principal ingredients. This involves outlining the rationale, objectives, and instructional sequence of one or more instructional computing programs. After this is outlined on paper, it is translated into the BASIC programming code. Following this, it is necessary to spend considerable time at a computer entering, testing, and refining what has been designed and developed on paper.

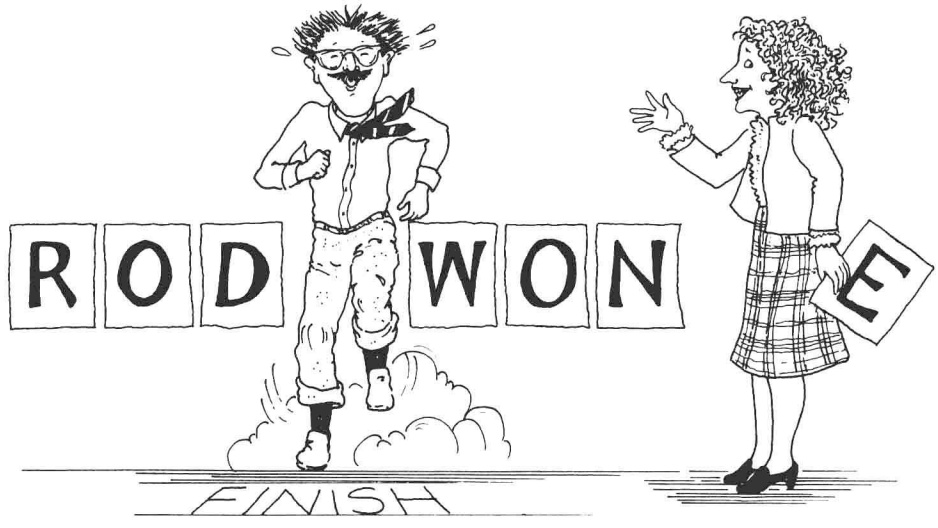
As a final introductory note, it should be emphasized that this book assumes no previous experience whatsoever with computers. On the other hand, it is not designed to provide detailed information on computers in general or how they operate. Rather, it introduces the ways and means by which the TRS-80 Model III or Model 4 microcomputer may be used within the instructional process.

Now, let us begin by getting down to the BASICs. . .

An Introduction to the BASIC Programming Language

"Nothing in life is to be feared. It is only to be understood."
—Marie Curie

"In certain trying circumstances, urgent circumstances, desperate circumstances, profanity furnishes a relief denied even to prayer."
—Mark Twain



Think About This (for Fun)

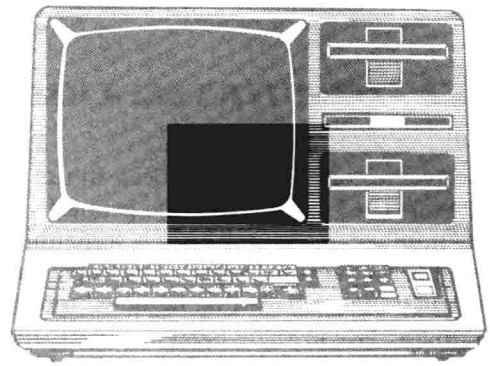
Rearrange the letters of NEW DOOR to form one word. [Note: Answers to *Think About This (for Fun)* Questions may be found in Appendix C.]

Think About This (Seriously)

Does a computer possess intelligence?

CHAPTER

1



A BASIC Program of My Very Own

1.1 OBJECTIVES:

For the successful completion of this chapter, you should be able to:

1. List five (5) general applications of instructional computing (Introduction).
2. Define two ways in which computers may be accessed (Section 1.3).
3. List the steps necessary to "boot up" a computer system (Appendix A).
4. State how a BASIC program may be entered on that system after the "booting up" (Section 1.6 and Appendix A).
5. Define what (not who) composes a BASIC program (Section 1.4.1).
6. Distinguish between BASIC statements and commands (Sections 1.4.1 – 1.4.2).
7. Define the action of the following BASIC commands: NEW, RUN, LIST, and SAVE (Section 1.4.2).

8. Define and give at least one example of both a NUMERIC variable and a STRING variable (Section 1.4.3).
9. Describe the use of commas and semicolons in BASIC for purposes other than punctuation (Section 1.4.4).
10. Define the purpose and give at least one example of the following BASIC statements: PRINT, INPUT, LET, and END (Sections 1.5.1 – 1.5.4).
11. Describe three simple techniques for editing BASIC programs (Section 1.6 and Appendix B, Section B.9).

1.2 COMPUTER USE: A BRIEF HISTORY AND RATIONALE

Electronic computers have been in use since the late 1940s. In the period from 1948 to 1965, they were used primarily for what their name implies: computing or "number crunching" as it is sometimes called. Starting about the mid-sixties, however, educators began experimenting with applications of computers in the instructional process that involved more than just computing.

In the decade following, this use expanded, and, just as computers have become ingrained in our society, instructional computing is becoming commonplace in our schools. (These points may be emphasized by the fact that since 1975 over 2,500,000 microcomputers have been purchased, many for home or school use.)

Now, it is very important to recognize that computers are not replacing teachers! The fundamental principle underlying the use of computers—regardless of the profession using them—is that they are an incredibly fast and accurate tool, and they allow people to do certain activities in a manner that heretofore was not possible. Thus, the use of computers in instruction is basically (no pun intended) that of *supplemental* applications. Computers allow teachers and students to do certain educational processes faster, with greater accuracy, and in a manner not possible before they came on the scene.

Computer programs can be very helpful in providing patient, routine drill on fundamental concepts, in generating and grading tests in a given discipline, and in many other applications. In any of these cases, the most effective programs are those designed by teachers, the professionals in the field who are aware of what is to be taught and how to teach it. As yet, there is no computer program that can lead an intelligent and sensitive discussion on any given abstract concept. There are no teachers out of a job because they have been replaced by a computer! That is something worth remembering.

1.3 ACCESS TO COMPUTERS

A computer is an extremely fast and accurate processor of data. In the simplest sense, most common computer systems may be viewed as four units connected electronically:

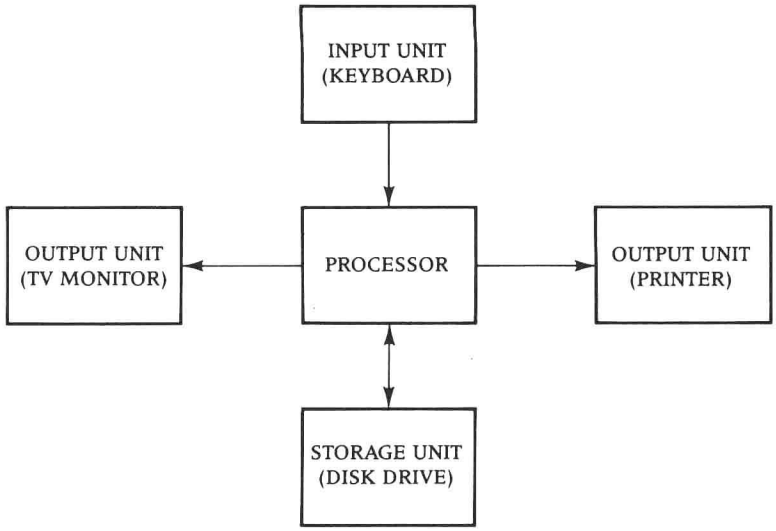


Figure 1.1
Component units of a computer system

1. An *input* unit (such as a computer terminal keyboard), through which data are entered.
2. A *processor* unit, which stores the data input and processes it electronically.
3. An *output* unit (such as a computer terminal screen or printer), which shows the results of processing the data input.
4. A *data storage/retrieval* unit (such as a disk drive), which stores data on, and retrieves data from, some magnetic medium (such as a floppy disk).

Figure 1.1 shows these units in block form.

Until the late 1960s, the primary means of access involved punching program statements, data, and commands onto computer cards. This "batch" of cards was read (input) by a card reader and eventually a printout (output) of the program "run" was retrieved. This type of access is commonly referred to as *batch* access or *batch* processing.

Since the early 1970s, there has been a very strong trend toward accessing computers via computer terminals. In the simplest sense, a terminal consists of a keyboard, similar to that of a typewriter, for input of statements, data, commands, and so forth, with output displayed either on a cathode ray tube (CRT) screen or paper (hardcopy) at the terminal. This type of access is known as *interactive* (a user is interacting directly with the computer or a program) or *timesharing* (there may be literally scores of terminals in remote locations "sharing the time" of one computer). In most instances, the terminal is connected to the computer via standard telephone lines.

Microcomputers are an exception to this. Here the computer, terminal, display, and other components are usually provided as a unit small enough to fit on a desk top (Figure 1.2). There are no telephone connections or sharing of computer

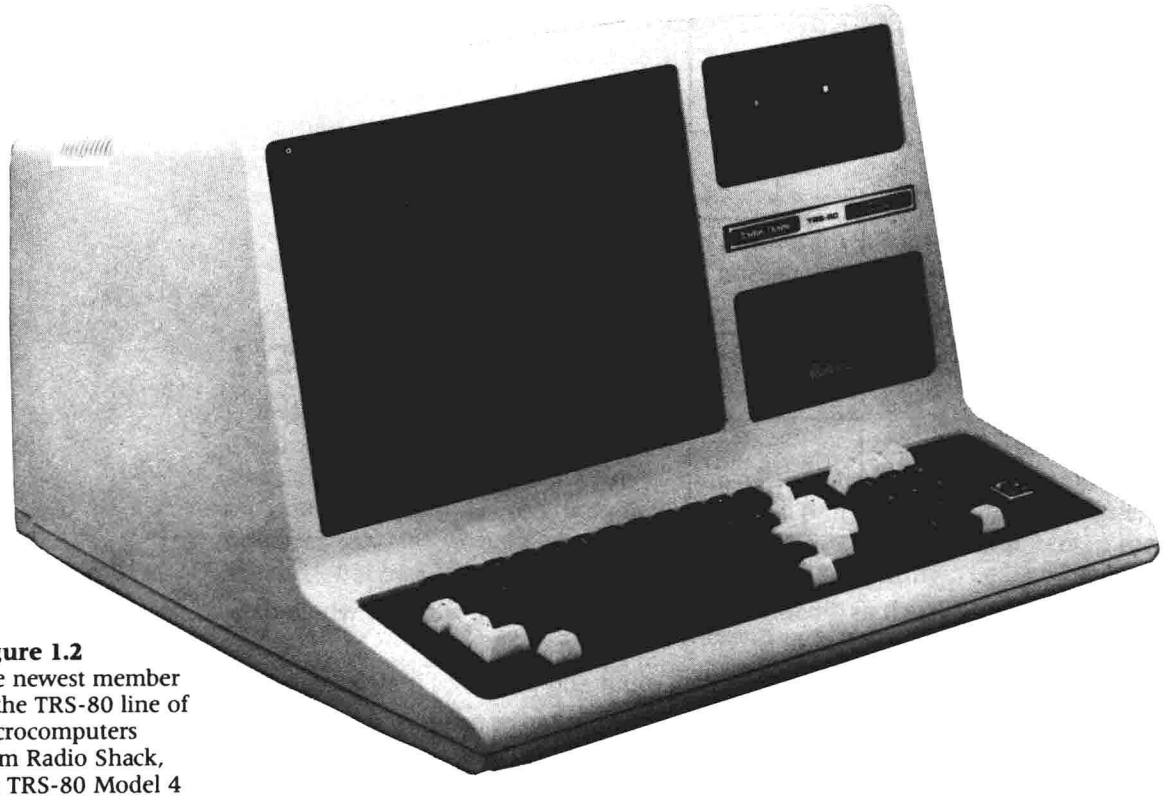


Figure 1.2
The newest member of the TRS-80 line of microcomputers from Radio Shack, the TRS-80 Model 4 Desk Top Microcomputer.

time. This makes the unit more portable, less prone to equipment failure, less expensive, and, consequently, well suited to the classroom.

For our use here, only microcomputers are discussed. The examples and assignments in the text assume that the reader has access to a TRS-80* Model III or Model 4 microcomputer with Disk BASIC, one floppy disk drive, a video monitor, and at least 48K of random access memory (RAM).

It is very important that the reader, particularly the reader new to microcomputers, become familiar with the processes needed to access (use) the system. This first involves gaining confidence in "booting up" the system. Refer to Appendix A for a step-by-step procedure to accomplish this.

1.4 A BIT ABOUT BASIC BEFORE BEGINNING

There are a few general points about BASIC that should be made early. Consider these as some of the "rules of the game" to follow for BASIC.

*TRS-80 is a registered trademark of Tandy Corporation.