AN APPLE® FOR THE TEACHER

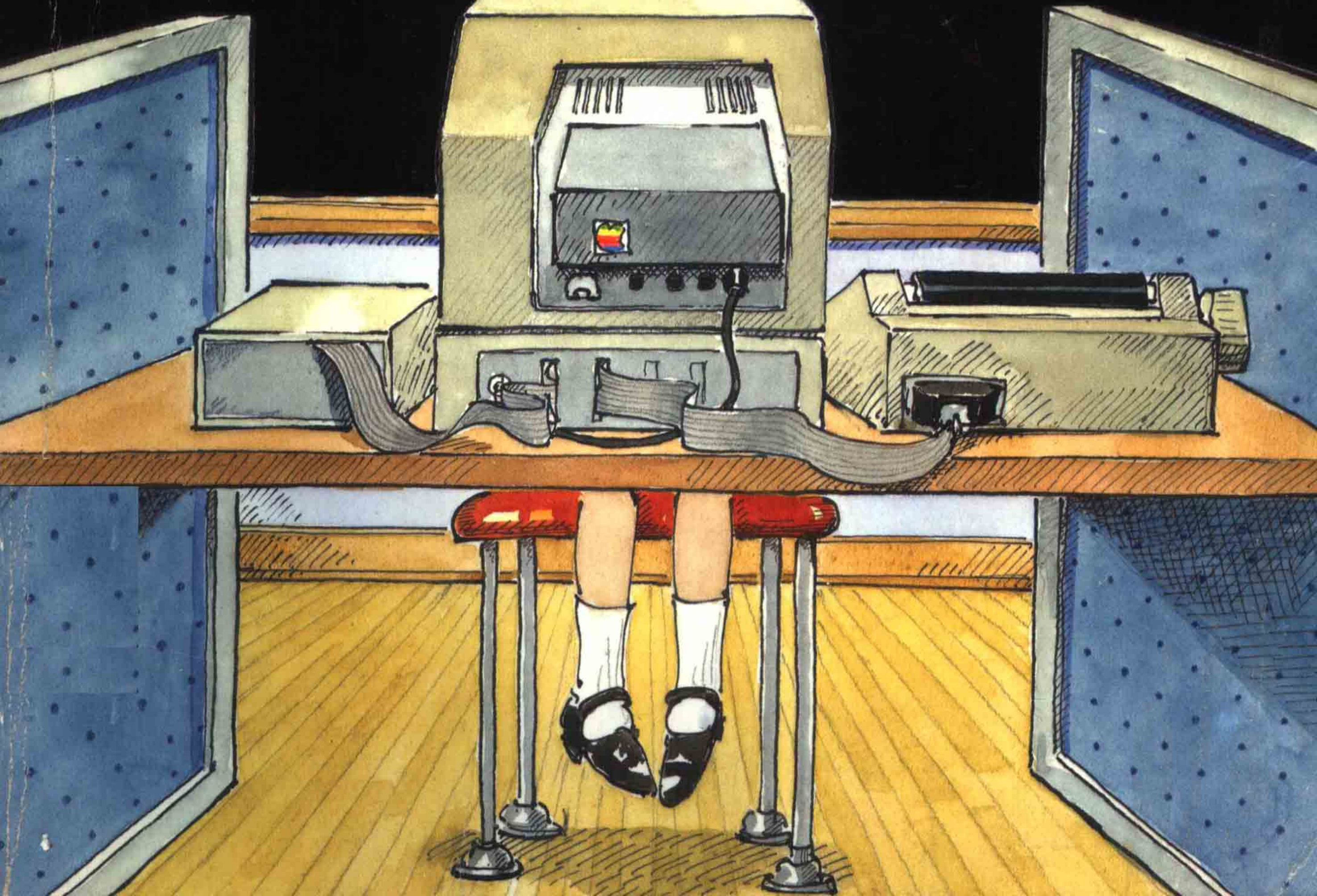
Fundamentals of Instructional Computing

Culp & Nickles

2nd Edition

NEW

```
10 HOME
20 VTAB 12: HTAB 18
30 PRINT "HOWDY"
```



AN APPLE FOR THE TEACHER

Fundamentals of Instructional Computing

SECOND EDITION

GEORGE H. CULP
University of Texas at Austin

RBERT L. NICKLES
California State University, San Bernardino

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

Culp, George H.
An Apple for the teacher.

(Brooks/Cole series in computer education)
Includes index.
1. Computer-assisted instruction—Teacher training.
2. Computer managed instruction—Teacher training.
3. Microcomputers—Teacher training. I. Nickles, Herbert. II. Title. III. Series.
LB1028.5.C78 1985 371.3'9445 85-30001

ISBN 0-534-05832-9

This product is neither endorsed nor coproduced by Apple Computer, Inc.
Apple II is a registered trademark of Apple Computer, Inc.

Sponsoring Editor: Neil Oatley
Editorial Assistant: Gabriele Bert
Production Service: Greg Hubit Bookworks
Manuscript Editor: Patricia Talbot Harris
Permissions Editor: Carline Haga
Interior Design: Rodelinde Albrecht
Cover Illustration: Andrew Myer
Typesetting: Vera Allen Composition
Printing and Binding: Malloy Lithographing, Inc.

PREFACE

This is a practical book for preservice and in-service teachers that presents the fundamentals of the BASIC programming language for the Apple II series of microcomputers and explains how to apply them to the design and development of instructional computing programs. In 15 years of teaching instructional computing courses to more than 2,000 students, it has been our experience that, given these fundamentals, teachers are able to expand on them and develop efficient programs to meet their specific needs.

The book focuses on the following elements:

- *Direct instructional computing application* The majority of programs designed and discussed may be easily adapted for actual classroom use.
- *Top-down design of programs* We believe that teachers can be introduced to structured design of programs, regardless of the programming language used. We incorporate this using techniques familiar to teachers who have been exposed to general instructional design strategies in their professional education.
- *Step-by-step solution for coding programs* Example programming "problems" incorporating the BASIC statements discussed in each chapter are solved in a step-by-step manner.
- *Frame-by-frame program format* Programs are designed in a frame (or screen) format, further emphasizing a top-down, structured method for development.
- *A systems approach to instructional design* Ten steps necessary for the design and development of instructional computing materials, along with suggested guidelines, are discussed in detail.

Although the book is directed to teachers who have little or no experience with computers, we want to emphasize that we are not attempting to teach general computer literacy. There is little mention of the history, architecture, or function in society of computers. Nor is this book for educating computer scientists or programmers. Several language statements common to programming texts are omitted because their application is not typical of the *instructional* use of computers. Rather, in keeping with our goal of providing fundamentals, we present what a teacher needs to know in order to begin designing anu developing instructional computing materials.

The book consists of ten chapters and five appendices, 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, tutorial dialog, simulation, and testing. Chapter 5 presents an introduction to string functions that are relevant to instructional computing applications. Chapters 6 and 7 give example programs in each of the five areas. In addition, Chapter 7 includes a series of teacher "utility programs" related to information storage and retrieval. With the exception of the simulation examples, each of the programs in these chapters may be easily modified or used directly in actual classroom applications. Chapter 8 discusses and demonstrates some simple uses of color and graphics as instructional techniques.

In Part Two, Chapters 9 and 10 discuss the specific steps needed to first design and then develop instructional computing materials. Ten criteria for evaluation of instructional computing programs are also included. The appendices include instructions for the general operation of the microcomputer and its disk operating system, a summary of the commands and statements and their uses, answers to selected questions and problems given in the chapters, instructions for making music with the Apple, and listings of programs given as solutions to certain problems.

As a matter of personal preference, some readers may wish to study Part Two on design and development before the chapters on BASIC. However, we believe that practical design and development can occur only after a working knowledge of the language is established. Thus, BASIC fundamentals are presented before instructional design. We want to emphasize again, however, that *program design* considerations are introduced early and followed throughout in Part One. We have found that presentation, demonstration, and discussion of the entire book requires 30 to 40 class contact hours. An equal or greater amount of student contact with the microcomputer is needed. Our general course format is to assign the design and development of one or more instructional computing units in the student's area of interest as a term project.

We discuss more than 30 programs related to instructional computing use. These programs are contained on a "text diskette" that is provided without cost by the publisher to adopters who order ten or more copies of the book. We believe that access to this disk is an *integral* and *important* consideration for those using the book. Therefore, we urge its use. Permission is given to copy the disk for class-related applications.

We extend our appreciation and acknowledgment to Carey Van Loon of California State University, San Bernardino, to Dee Dee Watkins and her husband, Morgan, of the University of Texas, Austin, and to Carla Cramer for her lasting or valuable enthusiasm. Special thanks go to Dennis Harper and Jeffrey Marcus, University of California, Santa Barbara; Sister Mary K. Keller, Clark College; Sylvia Pattison, University of Cincinnati; Frederick Schneider, Virginia Commonwealth University; Carl R. Stutzman, California State University, Fresno; Edward B. Wright, Western Oregon State College; and to the publication staff at Brooks/Cole Publishing Company. An extra measure of gratitude, and apologies for the burned steak, are given to our editor, Neil Oatley of Brooks/Cole.

CONTENTS

PROGRAMS

Examples Used in the Discussion of BASIC

Application Programs: Problem Solving

Application Programs: Drill and Practice

Application Programs: Tutorial Dialog

Application Programs: Simulation

Application Programs: Testing

Teacher Utility Programs Using Text Files

Graphics

Music

Listings of Answers to Posers and Problems Questions

Introduction

This book describes an approach to using a common programming language, BASIC, for the design and development of instructional computing programs for Apple II series microcomputers. Its ten chapters discuss certain fundamentals of BASIC and the design and developmental processes that provide a foundation for the production of instructional computing programs.

More than a hundred books are available that teach BASIC (the Beginner's 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 the language, they usually emphasize problem-solving applications. Our emphasis, on the other hand, is on instruction in the use of BASIC to design and develop materials for instructional computing.

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 logistic limits.
5. *Testing,* in which computer programs ask the questions, check the answers, and record the performance.

For our purposes, the term *instructional computing* includes 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 fundamentals needed to write simple programs for instructional computing applications.

Model programs are described that illustrate this use. Most of these programs may be easily modified for, or used directly with, actual classroom activities.

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 applications of instructional computing.
4. It is one of the introductory computer languages used in most secondary and many elementary schools.
5. It is the most common language of microcomputers—an area of computer technology making the major impact on education in this decade.

Top-Down 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. However, this general concept of a logical approach to program design is introduced in the early chapters of the book. Beginning with PROGRAM 3 in Chapter 2, each of the programs is designed in a "top-down," step-by-step, frame-by-frame approach. This is a familiar approach to instructional design for educators and facilitates program development, even for the novice computer user.

Development. The development of instructional computing programs by you 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 process involves outlining the rationale, objectives, and instructional sequence of one or more instructional computing programs. After this program design is outlined step by step, frame by frame on paper, it is translated into the BASIC programming code. The last steps are to spend considerable time at a computer entering, testing, refining, and evaluating what has been designed and developed on paper.

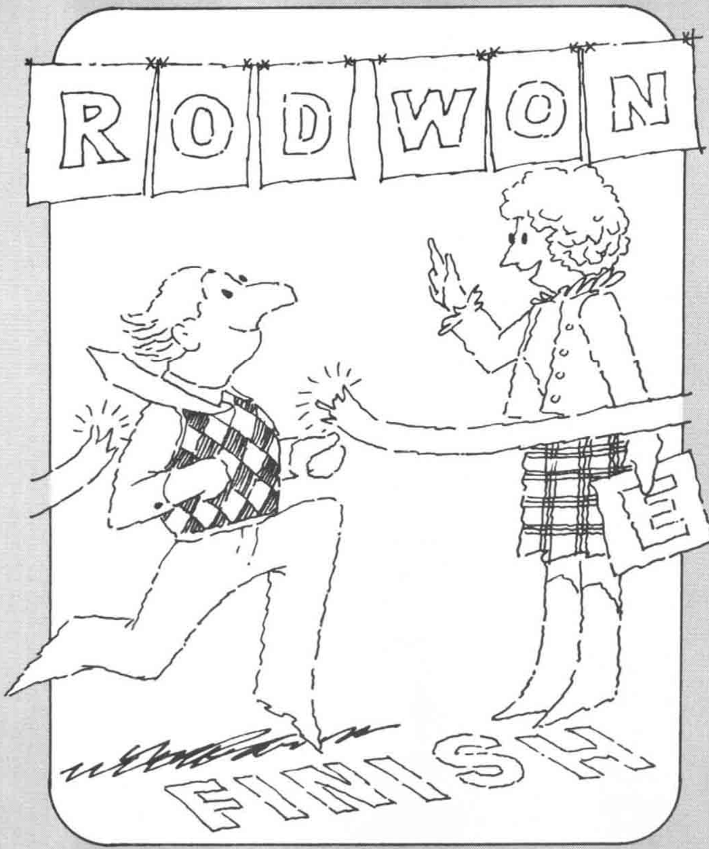
As a final introductory note, we emphasize that this book assumes no previous experience whatsoever with computers. On the other hand, the book does not provide detailed information on computers in general or how they operate. Rather, it introduces the ways and means by which the Apple II® series of microcomputers* may be used within the instructional process.

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

*Apple II is a registered trademark of Apple Computer, Inc.

PART ONE

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

"If at first you don't succeed, you are running about average."
—M. H. ALDERSON

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.)*

. . . 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 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" (power 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 actions 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 12,000,000 microcomputers have been purchased, many for home or school use.)

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 incredibly fast and accurate tools, and they allow people to do certain activities in a manner that has never before been 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.

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, we believe, 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 work 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:

1. An *input* unit (such as a computer terminal keyboard), through wnich data is 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.

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 may be considered an exception to this, although they may be connected to larger computers. 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 time. These features make 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 book assume that the reader has access to an Apple II series microcomputer with BASIC, one floppy disk drive, a video monitor or television, 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.