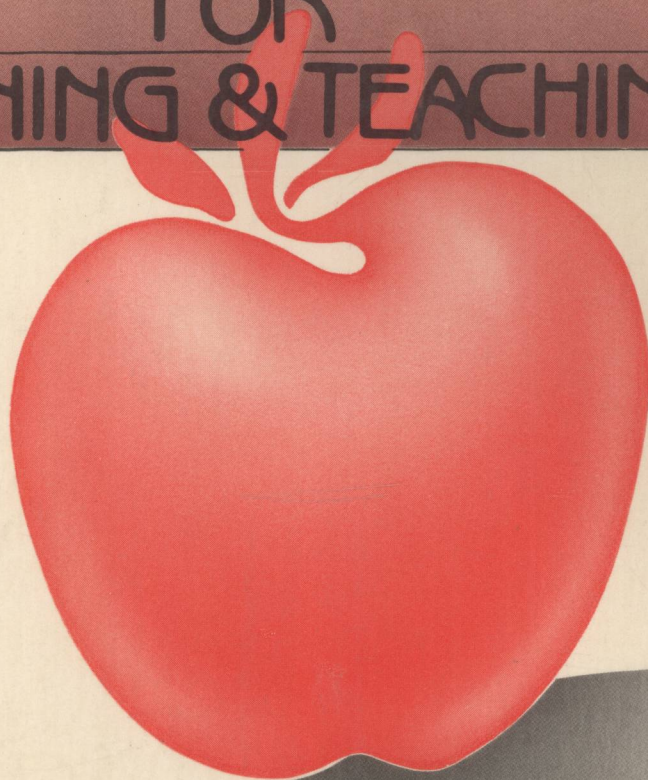


APPLE[®]
PROGRAMMING
FOR
LEARNING & TEACHING



50 + APPLICATIONS PROGRAMS

FREDERICK H. BELL

TP31
B27

8562530

APPLE® PROGRAMMING FOR LEARNING AND TEACHING

50+ Applications Programs

FREDERICK H. BELL

Professor and Coordinator
Educational Computer Laboratories
University of Pittsburgh



E8562530



A RESTON COMPUTER GROUP BOOK
RESTON PUBLISHING COMPANY, INC.
A Prentice-Hall Company
RESTON, VIRGINIA

This book is dedicated to my wife, Ellen Wainhouse Bell, who proofread the manuscript and served as my consulting editor, and to our children, Stephanie Alexis Bell and Jonathan Frederick Bell.

Library of Congress Cataloging in Publication Data

Bell, Frederick H.

Apple programming for learning and teaching : 50+ applications programs.

"A Reston Computer Group book."

Includes index.

1. Apple computer—Programming. 2. BASIC (Computer program language) 3. Computer programs. I. Title.

QA76.8.A66B45 1984 001.64'24 83-24781

ISBN 0-8359-0098-3

ISBN 0-8359-0097-5 (pbk.)

Apple® is a registered trademark of Apple Computer, Inc.

Editorial/production supervision and interior design
by Barbara J. Gardetto

© 1984 by

Reston Publishing Company, Inc.

A Prentice-Hall Company

Reston, Virginia 22090

All rights reserved. No part of this book may be reproduced in any way, or by any means, without permission in writing from the publisher.

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

APPLE®
PROGRAMMING
FOR LEARNING
AND TEACHING
50+ Applications Programs

Preface

Welcome to the fascinating world of personal computers. This book features more than 50 ready-to-use, teaching/learning programs and step-by-step instruction in the powerful, easy-to-learn Applesoft BASIC programming language. In addition to more than 50 applications programs, there are 80 smaller example programs to guide you in Applesoft BASIC, so that you can learn to write larger, professional-level programs. New programming commands, keywords, and techniques are described and demonstrated in these smaller programs. Then they are illustrated in larger applications programs using a building-block approach to learning Applesoft BASIC. Many programs are explained line by line so that you will understand what each program line does on the screen or in computer memory. This book can be used as a text in a BASIC programming and computer-applications course or as a self-study guide to learn BASIC on your own.

The “Introduction” deals with considerations in designing and developing different types of computer programs for learning and instruction.

Chapters begin with a “Chapter Preview” and end with a “Chap-

ter Summary.” A “Things to Do” section before each chapter summary contains chapter-review questions and activities. Answers to these review questions are given in Appendix C. Other teaching and learning aids include:

1. An expanded table of contents to assist you in locating applications programs, BASIC keywords and commands, and programming concepts throughout the book.
2. Screen-display layout charts for text and graphics characters, which are found in Chapters 4 and 10.
3. A summary of ASCII (American Standard Codes for Information Interchange) is presented in Chapter 8.
4. Chapter 12 contains a detailed summary with examples of BASIC commands, keywords, and conventions.
5. Appendix A is a glossary of computer terms used throughout this book.
6. A list of error-message codes, their meanings, and their probable causes is given in Appendix B.
7. A detailed index is an aid in locating programs and programming commands, keywords, and techniques.

Among the distinctive features of this book are:

1. An “Introduction” dealing with considerations for developing high-quality computer programs for teaching and learning
2. Sections on the nature and uses of personal computers
3. Well-documented example programs to make programming techniques and logic easy to understand and use
4. High-level applications programs to assist you in learning to write well-structured, professional-quality programs
5. An emphasis on programs for teaching and learning skills, concepts, and principles from various topics, either in school or at home
6. Illustrative programs with interactive games and animated graphics under user control for teaching and learning useful everyday skills.
7. Early introduction to graphics, FOR . . . NEXT loops, subscripted variables, READ/DATA statements, and GOSUB/RETURN to promote efficient, well-structured programming techniques

Although care has been taken in writing and testing the programs in this book on an APPLE II Plus computer, it is possible that

you might find a “bug” (error) in a program. Or something may not work exactly the same way on your computer as it does on mine. If this happens, check the *Applesoft BASIC Programming Reference Manual* that came with your computer, make appropriate changes for your computer, and examine the distinctive features of your own computer. A major part of using and learning to program computers is experimenting, correcting errors, and exploring how Applesoft BASIC and your own computer work. Debugging (correcting errors) is an important skill in becoming a good programmer.

This book was written for Applesoft BASIC users who have APPLE II, APPLE II Plus, or APPLE IIe computers.

APPLE, APPLE II, APPLE II Plus, APPLE IIe, and Applesoft are registered trademarks of Apple Computer, Inc., with regard to any computer product.

Frederick Bell

Contents



Preface xi

Introduction

DEVELOPING COMPUTER PROGRAMS FOR LEARNING AND TEACHING 2

- Considerations in Designing Teaching/Learning Programs 4
- Characteristics of Good Teaching/Learning Programs 6

Chapter 1

GETTING STARTED WITH YOUR PERSONAL COMPUTER 10

- What Is a Personal Computer? 11
- Some Capabilities and Limitations of Personal Computers 13
- Getting Started with Your Computer 14
- The Nature of Programs and Programming 17
- Lists and Runs of Ready-to-Use Programs 18
 - *Programs: SALE; SHAPES; TEMP*
- Things to Do 28
- Chapter Summary 28

Chapter 2

THE INS AND OUTS OF PROGRAMING 30

- Computer-Assisted Learning and Teaching 31
- Designing and Writing Well-Structured Programs 32
- Beginning BASIC Programming 34
- Lists and Runs of Ready-to-Use Programs 38
 - *Programs:* COUNT; CONFETTI; FLASH; SPELLBEAR
- Things to Do 44
- Chapter Summary 45

Chapter 3

PROGRAMMING TECHNIQUES FOR DISPLAYING INFORMATION 46

- A Review and Extension of BASIC Concepts 47
- Using Semicolons and Commas in Print Statements 50
- Powerful Programs with FOR/NEXT/STEP Loops 53
- Displaying Lines with PRINT TAB 57
- A Note About Correcting Program Statements 59
- Ready-to-Use Programs 59
 - *Programs:* INFLATE; ADD; CAPITALS
- Things to Do 64
- Chapter Summary 66

Chapter 4

PATTERNS AND PICTURES WITH RND, HTAB, VTAB, GR, HGR, PLOT, AND HPLOT 68

- Using REM, RND, INT, and Random Numbers 69
 - *Program:* MULTIPLICATION QUIZ
- Controlling the Screen with HTAB and VTAB 75
 - *Program:* A STARRY NIGHT
- Low-Resolution Colored Graphics 77
- High-Resolution Colored Graphics 85
 - *Program:* PLOTTING SHAPES
- Ready-to-Use Graphics Programs 90
 - *Programs:* AIRPLANE; LOW LINES; HIGH LASERS
- Things to Do 93
- Chapter Summary 95

Chapter 5

KEYWORDS FOR MORE FLEXIBLE AND EFFICIENT PROGRAMS 96

- Subscripted Variables and the Keyword DIM 97
 - *Program:* DIVISION QUIZ
- Conversational Programs with READ/DATA Statements 102
 - *Program:* STATES AND CAPITALS

More Power with GOSUB and RETURN	107
■ <i>Program:</i> SNOWY EVENING	
Ready-to-Use Applications Programs with Subroutines	110
■ <i>Programs:</i> CHUCK-A-LUCK; SOLVING PROPORTIONS	
Things to Do	116
Chapter Summary	117

Chapter 6

REVIEWING, EXTENDING, AND USING BASIC 118

Review and Extensions of Programming Techniques	119
■ <i>Programs:</i> ROUNDING NUMBERS; LENGTH IN CENTIMETERS; STORE INVENTORY	
Improving Programs with ON . . . GOTO Statements	129
■ <i>Program:</i> SQUARE ROOTS	
Using ON . . . GOSUB Statements	132
Putting It All Together in a Game Program	134
■ <i>Program:</i> DICE GAMES	
Editing Commands and Techniques	137
Things to Do	143
Chapter Summary	144

Chapter 7

ANIMATED GRAPHICS, LOGIC KEYWORDS, AND INTERACTIVE PROGRAMS 146

Making GR Graphics Pictures Move	147
■ <i>Programs:</i> JUMPING JACKS; DRIVING A TRUCK	
Logic Operations with the Keywords AND, OR, and NOT	154
■ <i>Programs:</i> MULTIPLY NEGATIVE NUMBERS; FIND THE UNICORN	
Direct Key Input and Timers with PEEK, POKE, CHR\$, and VAL	158
■ <i>Programs:</i> THE SCREEN AS A SKETCH PAD; A TIMED ADDITION QUIZ	
Things to Do	167
Chapter Summary	168

Chapter 8

FUNCTIONS AND TECHNIQUES FOR STRING AND SCREEN MANIPULATIONS 170

Using String Functions: LEFT\$, RIGHT\$, MID\$, and LEN	171
■ <i>Program:</i> COUNTRIES AND CAPITALS	
Using String Relations in Programs	177
A Program to Alphabetize Words	178
■ <i>Program:</i> ALPHABETIZE WORDS	

Using ASCII Number Codes with CHR\$ 180
STR\$, HTAB, VTAB, INVERSE, FLASH, NORMAL, and
SPEED 182

■ *Programs:* DIAGNOSTIC MULTIPLICATION;
SUBTRACTION QUIZ

A Few Notes About Applesoft BASIC 190

Things to Do 191

Chapter Summary 192

Chapter 9

ADVANCED MATHEMATICS: FUNCTIONS, GRAPHS, AND TECHNIQUES 194

Precision of Numbers and Order of Operations and Re-
lations 195

A Review and Extension of Variable Names 197

A Review of Order of Arithmetic Operations and Logical
Relations 199

Trigonometric Functions and Graphs 199

■ *Program:* GRAPH OF $Y = \sin(X)$

More Mathematics Graphs and Functions 203

■ *Programs:* GRAPH OF $Y = \log(X)$; GRAPH OF $Y = \exp(X)$; GRAPH OF AN ELLIPSE; GRAPH OF A CIRCLE; GRAPH OF A PARABOLA; FACTORIALS; A POLAR COORDINATE GAME

Defining Functions with DEF and FN 213

■ *Program:* COMPOUND INTEREST

Things to Do 218

Chapter Summary 219

Chapter 10

ONERR, RESUME, Data Arrays, More Graphics, and PEEK and POKE 220

ONERR GOTO, PEEK(222), and RESUME 221

■ *Program:* ERROR TRAPS

What to Do if the Computer Hangs Up 226

Data Arrays and Variables with Multiple Subscripts 227

■ *Programs:* IMPRESSIVE PHRASES; SALES DATA

Additional Features of Low- and High-Resolution Graphics 232

■ *Program:* A FULL-SCREEN ELLIPSE

Using Text Windows in the Text Screen Mode 237

■ *Program:* ARITHMETIC PROPERTIES IN A TEXT WINDOW

Things to Do 241

Chapter Summary 242

Chapter 11

MORE BASIC KEYWORDS AND THEIR USES 244

CLEAR, CONT, FRE, GET, POP, POS, STOP, SPC, TRACE, and
NOTRACE 245

More on PEEK and POKE 256

Special-Purpose Keywords 259

A Summary of Disk Operations 261

Conclusion 265

Things to Do 265

Chapter Summary 266

Chapter 12

A SUMMARY OF APPLESOFT BASIC 268

Special-Purpose Keyboard Symbols 270

Variable Names 271

Operators 271

Applesoft BASIC Commands 272

Functions 278

Boundaries and Ranges 279

APPENDICES 281

Glossary 283

Error Messages, Causes, and CODE Numbers 286

Answer Key 289

Index 299

APPLE®
PROGRAMMING
FOR LEARNING
AND TEACHING

50 + Applications Programs

INTRODUCTION

Developing Computer Programs for Learning and Teaching

The purpose of this book is to guide you in designing, developing, writing, and improving computer programs, especially programs for learning and teaching either in class or at home. The Applesoft® version of the easy-to-learn BASIC computer language, which is the most popular language on microcomputers, is used in this book. In addition to being used on microcomputers, BASIC is used on large multimillion-dollar computer systems, medium-size computers in the hundred-thousand dollar (and up) range, and on mini-computers priced in the range of tens of thousands of dollars.

You will learn to write professional-level, educational computer programs as you work through this book, which contains approximately 130 computer programs. These programs fall into three categories: (1) programs to illustrate specific programming techniques, (2) programs that illustrate both programming techniques and teaching/learning applications, and (3) 50 ready-to-use applica-

Applesoft® is a registered trademark of Apple Computer, Inc.

tions programs for teaching and learning a variety of useful school, college, and at-home skills.

CONSIDERATIONS IN DESIGNING TEACHING/LEARNING PROGRAMS

Most of the many thousands of teaching/learning programs for children, adolescents, or adults fall into eight overlapping categories according to their major purposes. These categories are:

1. *Programs to present facts and concepts.* Such programs present and give examples of new information, ideas, and concepts and describe what they mean and how they are used.

2. *Programs to explain and illustrate new skills.* These programs present and explore new skills that range from how to use a microcomputer, to learning typing, through how to do a variety of mathematics computations or improve reading comprehension.

3. *Programs for practicing skills.* Skills such as reading, spelling, typing, arithmetic computations, grammar, and even programming computers require lots of practice. Computer programs that are interactive and flexible have been developed to assist people in practicing a large variety of skills.

4. *Programs for interactive tutoring.* As people learn facts, skills, concepts, and principles, some things are learned and practiced incorrectly. If “practice makes perfect,” practicing things incorrectly leads to perfection at doing things wrong. Once a person learns and practices a skill incorrectly, it takes much extra practice to “unlearn” incorrect patterns in preparation for relearning them correctly. Many instructional-computing programs are designed to tutor learners in understanding and correctly learning facts, skills, concepts, and principles and to help them avoid misconceptions and error patterns.

5. *Tests and quizzes.* One measure and evaluation of learning is performance on tests and quizzes. Some computer programs are designed to give students tests and quizzes. These programs immediately evaluate each of a student’s responses so that he or she will not repeat and reinforce incorrect answers and procedures.

6. *Decision making* is an important skill in our society. Many computer programs are designed to simulate “real-world” situations where decisions must be made. Such programs may simulate an experiment in physics or chemistry, put the user in control of a nuclear power plant, or require one to read a map correctly to reach a specified destination quickly. During simulations and games, situations and problems occur that require decisions to be made by the computer user. The result of each decision or the consequences of a

pattern of decisions are communicated to the user. Through experience using the simulation program, the person practices decision making and learns to make decisions that cause desirable results and to avoid decisions that result in undesirable outcomes.

7. Creativity. Creativity is difficult to “teach” and hard to evaluate using objective standards. A good way to “teach” creativity is to encourage people to involve themselves in creative acts and provide advice and constructive criticisms of their creative efforts and products such as dancing, painting, woodworking, writing, and even writing computer programs. Computer programs have been developed to assist in some of the drudge work in producing creative works such as architectural plans, colored patterns, animated films, and creative writing. Writing computer programs to display computer art, colored patterns, stories and poems, and animated graphics (moving pictures) is a good way to practice creative activities.

8. Problem solving. Solving unique or distinctive problems is one of the highest-level activities of people. Problem solving, which is also a creative act, involves several stages. They are:

- a. An idea for or awareness of a situation (problem) that needs to be resolved
- b. Evaluating the idea and formulating it into a coherent, organized, well-structured statement and plan
- c. Gathering, sorting, and assessing information and procedures that may be useful in resolving the situation (solving the problem)
- d. Developing solution strategies and procedures and testing them
- e. Arriving at a solution, or alternative solutions, to the problem
- f. Evaluating a solution or alternative solutions to determine the appropriateness or acceptability of solutions
- g. Assessing and modifying the problem-solving process and procedures so that problem-solving techniques and strategies can be generalized to other problem-solving situations

Appropriately designed-and-developed computer programs can promote practicing and improving problem-solving abilities. It is interesting to observe that designing and writing computer programs is itself a high-level problem-solving activity. Writing a program includes:

- Formulating an idea as to what the program should do
- Developing a well-structured plan
- Organizing procedures for implementing the plan
- Writing the program