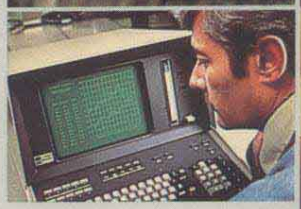
PROGRAMMING IN BASIC

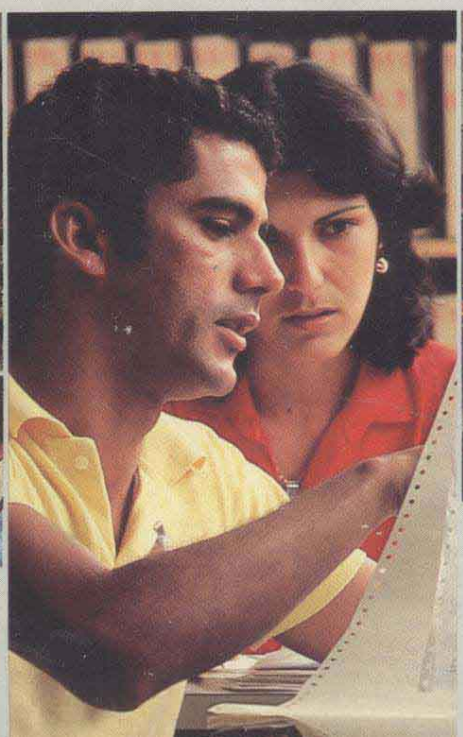
Ralph M. Stair, Jr

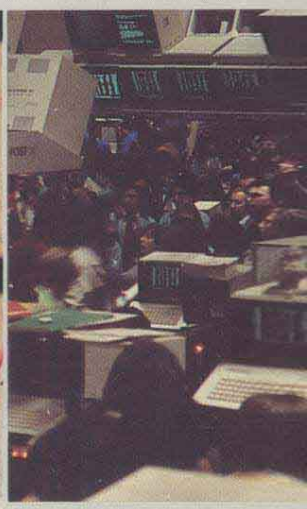
Structured Programming,

Cases, Applications,

and Modules







Programming in BASIC

With Structured Programming, Cases, Applications, and Modules

Ralph M. Stair, Jr.
The Florida State University

Third Edition

1985

RICHARD D. IRWIN, INC. Homewood, Illinois 60430

Cover photos courtesy of (left to right):

Honeywell Information Systems
Tektronix
Bell Laboratories
IBM Corporation
Chicago Board Options Exchange

ISBN 0-256-03213-0

Library of Congress Catalog Card No. 84–81735

Printed in the United States of America

1 2 3 4 5 6 7 8 9 0 ML 2 1 0 9 8 7 6 5

Dow Jones-Irwin, a division of Richard D. Irwin, Inc., has developed Personal Learning Aids (PLAIDS) to accompany texts in this subject area. Copies can be purchased through your bookstore or by writing PLAIDS, 1820 Ridge Road, Homewood, IL 60430.

Programming in BASIC

With Structured Programming, Cases,
Applications, and Modules

The Irwin Series in Information and Decision Sciences

To Lester and Lola Harkrider

Acknowledgments

The successful completion of any text requires a team effort, and there were many individuals and organizations that contributed to this project.

This book has been dedicated to two individuals who have molded my life and this project in more ways than one. Their own success has been an inspiration. Lester started his career as the office messenger and retired as chairman of the board. At the age of 82, Lola wrote her first book.

The management area and the College of Business at The Florida State University continue to give their support for this and other projects.

Ralph Janaro, Steve Traczyk, Thomas O'Connor, Charles Langdon, Mohan Mankikar, Mary Dovie, and Brian Blowers have also been very helpful in developing this book. Furthermore, I would like to thank Dan Juhn, the Department of Management and Marketing, and the College of Business Administration at the University of New Orleans for their support and encouragement. Barry Render and Mickey LeBoeuf also provided friendship and encouragement.

Gary Wicklund, Connie McLaren, Claude McMillian, Robert Fetter, and Beau Loughlin provided me with some very helpful suggestions and comments concerning the project.

Thanks are also due to the reviewers of the manuscript: George H. Evinger, Azusa Pacific University; Eugene F. Stafford, Iona College; Eugene L. VandenBoss, Adrian College; William R. Kenney, San Diego Mesa College; and, Larry Seiford, The University of Texas at Austin.

To the Instructor

This text is designed to introduce students to the BASIC programming language. The approach has been to make the material and presentation useful to the student in other courses and during the student's professional career. In addition, this book has been designed to allow you greater flexibility in assigning course material. Thus this book can be used when the BASIC language is to be covered along with other material. With increasing class sizes and more demands on your time, this book has also been designed to make the dissemination of material and the evaluation of student performance as easy and as straightforward as possible.

Each chapter has been constructed to make it easier for students to grasp the material and to assess their performance on the material before proceeding to new material. Several *learning objectives* appear at the beginning of each chapter. Each objective has been written to be clear and measurable. This makes it easier for a student to determine whether or not the material has been mastered. Two or more *self-quizzes* are included in each chapter. When the material builds and a previously learned concept or statement is needed to fully understand new concepts and statements, it is important for students to determine their weaknesses and to correct these weaknesses or deficiencies before they start on new material. *Questions* and *exercises* at the end of each chapter further test the student's ability to understand and use the material and the degree to which the student obtained the learning objectives presented at the beginning of each chapter. Each question is designed to test a student's understanding of a concept or statement. The exercise normally requires one or more BASIC programs to be written to solve a particular problem. The degree of difficulty increases with each question or exercise. For your convenience, the exercises have been divided into "exercises" and "advanced exercises." The degree of difficulty of the exercises and questions range from almost trivial to extremely complex. Thus the first few exercises in each chapter may only require three or four statements, and the last exercises may require a substantial amount of work. Of course, the other exercises are moderately difficult. This type of arrangement is especially useful when you allow students to attempt to solve more complex problems for extra credit. Finally, a performance checklist

appears at the end of each chapter. This checklist relates each self-quiz, question, and exercise to the learning objectives stated at the beginning of each chapter. Thus a student is able to measure the degree to which he or she has been able to accomplish the stated learning objective.

To make the material more relevant to the student in other courses and during the student's professional career, each chapter contains several *applications.* Each application investigates a problem that requires one or more BASIC programs as part of the solution. These applications test the student's understanding of a few concepts or statements, reveal to the student how these statements or concepts can be used in writing BASIC programs, and show how BASIC programs can be used to solve problems for organizations that are in both the private and public sector. In addition, two chapters have been included that present a variety of *cases* to allow students to use all of the statements and concepts learned. These chapters permit students to solve problems for different types of organizations by writing one or more BASIC programs. Containing more than 20 cases total, these chapters give students an opportunity to pull all of their programming skills and knowledge together in solving realistic and useful case problems. Like the exercises at the end of each chapter, the cases increase in degree of difficulty. In fact, the cases have been divided into "cases" and "advanced cases." Some of the cases only require several short and simple statements. Other cases are more complex. One case requires the development of a computer simulation, and another case requires six separate computer programs interacting with five different data files to prepare a realistic payroll application that prepares checks, change reports, error reports, master file listings, exception reports, and summary reports. Most of the cases are between these two extremes, providing the average student with the opportunity to solve problems for organizations in the public and private sector by writing realistic and useful BASIC programs.

A great deal of flexibility has been achieved by dividing this book into *three parts.* Many instructors who teach BASIC programming first in a course that includes other material find that the books they have used do not do an adequate job of introducing the student to computers and data processing. Chapter 1 of this book introduces students to some fundamental concepts of computers and data processing. Chapter 2 and Chapter 3 investigate writing successful computerized applications and the human side of writing such applications. Thus, if BASIC programming is the first subject to be taught in a course that also covers other material, Part 1 may be used to introduce students to computers, data processing, and computer programming. Part 2 contains fundamental statements and concepts of BASIC. LET, IF, READ, DATA, GO TO, FOR and NEXT, INPUT, subscripted variables, and so on. Part 3 contains additional features of BASIC. GO SUB, DEF, RND, PRINT USING, TAB, data files, MAT statements, and so forth. If you want only to introduce students to the fundamental concepts of BASIC, Part 2

should be covered. If you want to emphasize BASIC, then Part 3 should be covered. Of course, if your course is between these two approaches, you can cover Part 2 and selected chapters or concepts from Part 3.

To provide even greater flexibility, several modules have been included that can be covered at the instructor's discretion. These modules cover locating and correcting errors, structured programming and design, extended and enhanced BASIC statements, and microcomputers for business, home, and school use.

Increasing class sizes, reduced resources, and more demands on your time make it important for any text to provide the instructor with a package that makes course design and student evaluation easier. This is one of the only BASIC programming texts that has a *complete and comprehensive instructor's manual*. Helpful *hints* and suggestions have been included to assist you in designing the course and preparing course outlines and related material. To help you prepare for your lectures, *chapter outlines and notes* have been included to assist you in your presentation. Furthermore, all of the transparency masters have been keyed to the chapter outlines and notes. The *answers to all questions and exercises* have been included in the instructor's manual, and there are close to 3,000 *true-false and multiple-choice test questions*.

This text was written using electronic manuscript preparation. The entire text was entered onto floppy disks using a word processing program and several microcomputers. Then, a spell check program was used, and the manuscript was extensively edited. After this, we went directly from the floppy disks to the final pages. Without electronic manuscript preparation, most books are entered into the typesetter's computer by a data entry operator. This takes more time, and there is a greater likelihood for errors and problems. Electronic manuscript preparation cannot eliminate all errors, but it can significantly reduce the number of errors and potential problems.

A tremendous amount of progress has been made in technology and the techniques that are used in developing computer programs. Much of this progress has been after the original development of the BASIC programming language. As a result, BASIC is not an ideal language to use concepts such as structured and modular design. Although some computer systems have developed more structured statements, other computer systems have not. Whenever possible, we have attempted to use the concepts of structured design, modular design, and good internal documentation. In addition to teaching students how to write programs that work, we also stress good programming style.

This text has been designed to be relevant to the student and convenient for you. Many features have been incorporated into this package to accomplish these objectives. Any comments you have that would help to strengthen this package would be greatly appreciated.

R. M. S.

To the Student

Today, a complete computer system can be purchased for less than $100 or for more than $1 million. Students are purchasing microcomputers or personal computers for schoolwork, and the largest companies and organizations are purchasing large computers for more efficient and effective operations. Our society and way of life is dependent on the computer. There is no way of going back to the old manual method of processing data; computers will be with us forever.

The BASIC programming language is an excellent language to learn. Most small computers and microcomputers use the BASIC programming language. Most larger computer systems have the ability to use BASIC as well as other popular languages, such as COBOL and FORTRAN. As computer systems become less expensive, BASIC will become more popular. The BASIC programming language is easy to learn. It also has some features and built-in conveniences that make it very effective and very powerful.

The objective of most books on BASIC is to teach students how to program in the BASIC language. This objective, however, may not be sufficient if the programs that you learn to write are trivial, unrealistic, or irrelevant to your needs. In this book you will learn how to program in the BASIC language. But this book has also been designed to be helpful to you as a student and as a professional in your chosen field. Furthermore, the topic coverage and progression has been designed to make the material straightforward and meaningful. Each statement that is discussed will allow you to write programs with less effort or to write programs that would be difficult or impossible without the particular statement or concept.

To help you monitor your progress through this book, *learning objectives* are stated before each chapter, and a *checklist* is included at the end of each chapter to help you assess your progress. There are several *self-quizzes* in each chapter. Each self-quiz has been developed to test your knowledge and skills on concepts and statements before you start studying more material. The chapters also contain several *applications* that reveal the use of statements and concepts in solving problems for a variety of

Microcomputer Differences

Today, most microcomputers use BASIC as their primary language. Unfortunately, many of the statements vary slightly from one microcomputer to another. Most chapters reveal some of these differences for popular statements and concepts. A separate section in each chapter, entitled "Microcomputer Differences," will show the differences between what is discussed in the book and what works for popular microcomputers. If the statements are the same, you will see the words *the same*. If there are no differences, but important enhancements exist, these enhancements will be indicated. Because of space limitations that exist for any book, the text concentrates on major statements and concepts. Most microcomputers have a wide range of statements beyond the fundamental statements. If you are using a microcomputer, you may wish to refer to the BASIC reference manual. For examinations, unless told otherwise by your instructor, use the material and general forms discussed in the book and not the concepts for a particular microcomputer.

Contents