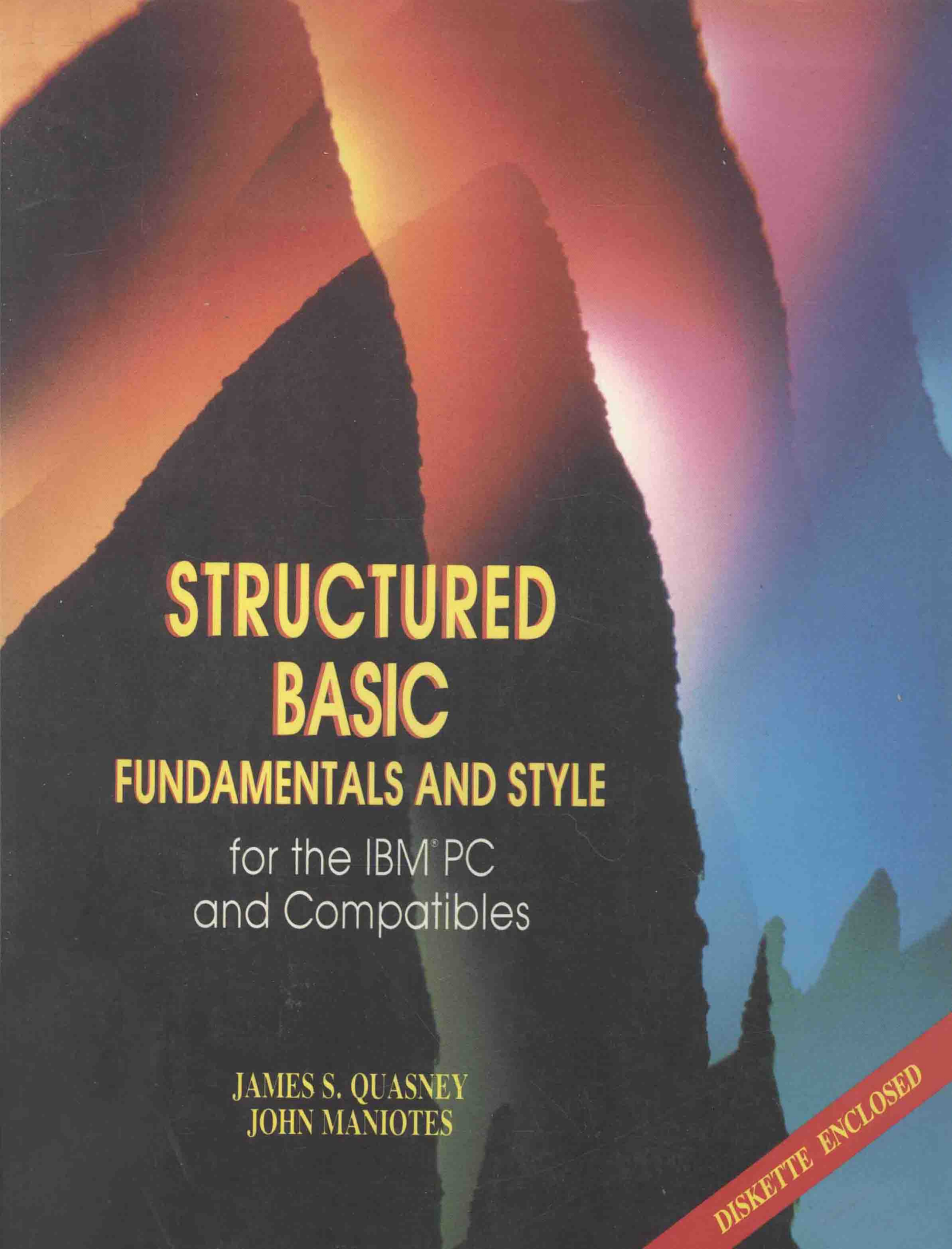

STRUCTURED
BASIC
FUNDAMENTALS AND STYLE
for the IBM® PC
and Compatibles
JAMES S. QUASNEY
JOHN MANIOTES
DISKETTE ENCLOSED

STRUCTURED BASIC

FUNDAMENTALS AND STYLE

for the IBM® PC
and Compatibles

James S. Quasney
John Maniotes

Purdue University Calumet

BOYD & FRASER PUBLISHING COMPANY
BOSTON

Dedication

To our wives: ***Linda*** and ***Mary***
The Quasney tribe: ***Lisa***, ***Jeff***, ***Marci***, ***Jodi***, ***Amanda***, and ***Nikole***
The Maniotes clan: ***Dionne***, ***Sam***, and ***Andrew***

Credits:

Publisher: Tom Walker
Editor: Donna Villanucci
Director of Production: Becky Herrington
Design/Cover: Becky Herrington
Cover Photo: ©David Hughes/Stock Boston
Manufacturing Director: Erek Smith

4 5 6 7 8 9 K 5 4 3 2 1

Manufactured in the United States of America

Library of Congress Cataloging-in-Publication Data

```
Quasney, James S.
    Structured BASIC fundamentals and style / James S. Quasney, John
  Maniotes.
       p.   cm.
    Includes index.
    ISBN 0-87835-289-9
    1. BASIC (Computer program language)  2. Structured programming.
  I. Maniotes, John.  II. Title.
  QA76.73.B3Q376 1988
  005.13'3--dc19                                              87-29968
                                                                CIP
```

STRUCTURED BASIC

FUNDAMENTALS AND STYLE

for the IBM® PC
and Compatibles

Programming books from boyd & fraser

Structuring Programs in Microsoft BASIC
BASIC Fundamentals and Style
Applesoft BASIC Fundamentals and Style
Complete BASIC: For the Short Course
Fundamentals of Structured COBOL
Advanced Structured COBOL: Batch and Interactive
Comprehensive Structured COBOL
Pascal
WATFIV-S Fundamentals and Style
VAX Fortran
Fortran 77 Fundamentals and Style
Learning Computer Programming: Structured Logic, Algorithms, and Flowcharting
Structured BASIC Fundamentals and Style for the IBM® PC and Compatibles
C Programming
dBASE III PLUS Programming

Also available from boyd & fraser

Database Systems: Management and Design
Using Pascal: An Introduction to Computer Science I
Using Modula-2: An Introduction to Computer Science I
Data Abstraction and Structures: An Introduction to Computer Science II
Fundamentals of Systems Analysis with Application Design
Data Communications for Business
Data Communications Software Design
Microcomputer Applications: Using Small Systems Software
The Art of Using Computers
Using Microcomputers: A Hands-On Introduction
A Practical Approach to Operating Systems
Microcomputer Database Management Using dBASE III PLUS
Microcomputer Database Management Using R:BASE System V
Office Automation: An Information Systems Approach
Microcomputer Applications: Using Small Systems Software, 2/e
Mastering Lotus 1-2-3
Using Enable: An Introduction to Integrated Software
PC-DOS Simplified

Shelly, Cashman, and Forsythe books from boyd & fraser

Computer Fundamentals with Application Software
Workbook and Study Guide to accompany Computer Fundamentals with Application Software
Learning to Use SUPERCALC®3, dBASE III®, and WORDSTAR® 3.3 An Introduction
Learning to Use SUPERCALC®3: An Introduction
Learning to Use dBASE III®: An Introduction
Learning to Use WORDSTAR® 3.3: An Introduction
BASIC Programming for the IBM® Personal Computer
Workbook and Study Guide to accompany BASIC Programming for the IBM® Personal Computer
Structured COBOL — Flowchart Edition
Structured COBOL — Pseudocode Edition
Turbo Pascal Programming

PREFACE

OBJECTIVES OF THIS BOOK

This book was developed specifically for an introductory computer programming course that utilizes Microsoft BASIC on an IBM PC, IBM PS/2, or compatible system. The objectives of this book are as follows:

1. To acquaint the reader with the proper and correct way to design and write high-quality programs. The top-down approach and structured programming are emphasized early and are consistently used throughout the book. The `GOTO` statement is not used in any program examples.
2. To teach the fundamentals of the Microsoft BASIC programming language.
3. To familiarize the reader with the operational procedures of the IBM PC, IBM PS/2, and compatible systems.
4. To teach good problem-solving techniques that can be used in advanced computing and information processing courses.
5. To emphasize interactive applications and menu-driven programs, the most popular type of programming in today's world.
6. To develop an exercise-oriented approach that allows the reader to learn by example.
7. To use practical problems to illustrate the applications of computers.
8. To encourage independent study and help those who are working alone on their own personal computer systems.

LEVEL OF INSTRUCTION

This book is designed to be used in a one-semester course in BASIC programming or in independent study by those who are working alone. No previous experience with a computer is assumed, and no mathematics beyond the high school freshman level is required. The book is written specifically for the student with average ability, for whom continuity, simplicity, and practicality are characteristics we consider essential. Numerous insights, based on the authors' fifty cumulative years of experience in teaching and consulting in the field of data processing, are implicit throughout the book. For the past ten consecutive years, we have both taught introductory programming courses using BASIC.

FUNDAMENTAL TOPICS ARE PRESENTED IN DETAIL

Besides introducing students to the proper and correct way to design and write programs by means of structured and top-down techniques, this book presents fundamental topics concerning computers and programming which should be covered in any introductory programming class. These include the stored program concept; getting on the computer; editing programs; input/output operations; variables and constants; simple and complex computations; the use of functions and subroutines; decision making; the use of counters and running totals; rounding and truncation; looping and end-of-file tests; counter-controlled loops; the use of logical operators; string manipulation; and graphics. Other essential topics

include data validation; control breaks; paging reports; table processing; sequence checking; selection; searching; matching; merging; sorting; file processing; and the differences between batch and interactive applications. Every one of these topics is covered in detail in this book.

DISTINGUISHING FEATURES

The distinguishing features of this book include the following:

A Proven Book

This book has evolved over the past decade and is based on the authors' four prior books on BASIC programming. Many instructors who have used our books have shared with us their comments and suggestions for improvement as new programming techniques have been developed. They have done much to shape the contents of this book, which reflects modern programming practices.

Early Presentation of the Top-Down (Modular) Approach and the Structured Programming Approach

Students are introduced to the top-down approach early, before they learn about looping and decision making. By the time they get to the larger and more complex programs, they are solving problems top-down by habit.

Particular attention is given to designing proper programs by means of the three logic structures of structured programming: Sequence, Selection (If-Then-Else and Case), and Repetition (Do-While and Do-Until). A disciplined method for implementing the structured design is adhered to throughout the book.

GOTOless Textbook

The `GOTO` and `ON-GOTO` statements are discussed briefly. However, they are not used in this book to write programs. All looping is implemented by means of either the `WHILE` and `WEND` statements or the `FOR` and `NEXT` statements. `GOSUB` statements are used when `IF` statements require more than three physical lines.

Early and Complete Coverage of File Processing

Complete coverage of sequential, random, and simulated-indexed files provides the reader with knowledge that is central to a real programming environment. Topics include creating all three types of files; file maintenance (matching and merging operations) and an information retrieval system that features simulated-indexed files. Sequential file processing is covered immediately following the presentation of the top-down approach and structured programming.

Student Interaction With Programs on the Student Diskette

The Student Diskette that accompanies this book contains all the executable programs and data files. Following the discussion of a program, the student is often asked to load the program from the Student Diskette and alter one or more statements to generate new results. We use the heading Try It Yourself to signal this activity. Instructors are encouraged to add to these short but meaningful exercises.

In our opinion, this interaction with the program has great educational value. Students quickly realize the significance of a statement by reviewing the results due to its modification. Interacting with a program also allows the student to experiment with changes without having to type the entire program. Furthermore, students may use a program from a particular chapter as a shell to solve a problem that is presented at the end of that chapter.

The use of a shell program will save keying time. Finally, the instructor will not be deluged with students requesting copies of data files when exercises requiring them are assigned, since each student will already have his or her own personal copy of all the referenced data files.

BASIC Programming Problems with Sample Input and Output

A total of 60 challenging field-tested BASIC Programming Problems are included at the end of the chapters. Each of the 60 problems includes a statement of purpose, a problem statement, sample input data, and the corresponding output results. Solutions to these problems are given in the *Instructor's Manual and Answer Book* and are also available from the publisher on an IBM PC-compatible diskette.

Interactive Applications (Menu-Driven Programs)

Although examples of batch processing are presented, the primary emphasis is on interactive processing. The reader is introduced to the `INPUT`, `PRINT`, and `CLS` (Clear Screen) statements early in Chapter 2. The `LOCATE` statement is presented in Chapter 4 and thereafter is used extensively to build screens. Several menu-driven programs are illustrated to familiarize the reader with the type of programming that is proliferating today.

Emphasis on the Program Development Cycle

The program development cycle is presented early in Chapter 1 and is used throughout the book. Good design habits are reinforced, and special attention is given to testing the design *before* attempting to implement the logic in a program.

Emphasis on Fundamentals and Style

Heavy emphasis is placed on the fundamentals of producing well-written and readable programs. A disciplined style is consistently used in all program examples. Thorough documentation and indentation standards illuminate the implementation of the Selection and Repetition logic structures.

Summary of the Microsoft BASIC Language on a Reference Card

A summary of the statements, commands, functions, special variables, special keys, operators, and reserved words can be found on a reference card at the back of this book. This summary is invaluable to the beginning student as a quick reference piece.

Presentation of Programming Case Studies

This book contains 26 completely solved and annotated case studies, illuminating the use of Microsoft BASIC and personal computers in the real world. Emphasis is placed on problem analysis, program design, and an in-depth discussion of the program solution. The program solutions to these Programming Case Studies, as well as all other programs found throughout the book, are on the accompanying Student Diskette.

Program Design Aids

Since the authors recognize top-down charts and flowcharts as excellent pedagogical aids and as the tools of an analyst or programmer, many of the Programming Case Studies include both top-down charts and program flowcharts to demonstrate programming style, design, and documentation. For the student's convenience, line numbers have been placed at the top-left corner of the symbols to better illustrate the relationship between the logic diagrams and the program.

Debugging Techniques and Programming Tips

A characteristic of a good programmer is that he or she has confidence that a program will work the first time it is executed. This confidence implies that careful attention has been given to the design and that the design has been fully tested. Still, errors do occur, and when they do, they must be corrected. Throughout this book, especially in Appendix C, efficient methods for locating and correcting errors are introduced. Tracing, as well as other debugging techniques, is discussed in detail. The section in Appendix C which deals with programming tips serves as an excellent reference, facilitating the writing of efficient, readable code.

Applications-Oriented Approach

Over 150 Microsoft BASIC programs, illustrating a wide range of practical applications, along with many partial programs, are used to introduce specific statements and the proper and correct way to write high-quality programs.

Emphasis on Data Validation

Most abnormal terminations in a production environment are due to user errors and not programmer errors. This is especially true for programs that interact with the user or are executed on personal computers. Good programmers will attempt to trap as many user errors as possible. This book pays particular attention to the illustration of various data validation methods for ensuring that incoming data is reasonable or within limits.

What You Should Know

Each chapter contains a succinct, list-formatted review entitled What You Should Know, reinforcing key concepts and data processing terminology.

Test Your BASIC Skills

A set of short-answer exercises identified as Test Your BASIC Skills, appears at the end of each chapter. Over 200 problems, many of which are complete programs, are included for practice. Through the use of these exercises, students can master the concepts presented and instructors are afforded a valuable diagnostic tool. Answers for even-numbered Test Your BASIC Skills exercises are included at the end of the book, before the index. Answers to the odd-numbered exercises can be found in the *Instructor's Manual and Answer Book*.

Graphics and Sound

Chapter 10 covers all the graphics statements and functions in Microsoft BASIC that are central to understanding what can be done with graphics on the PC. The topics provide the student with knowledge of how to create, change, display, and store graphic designs and animation sequences. Furthermore, the necessary sound and music statements are discussed and are applied to various applications.

Additional PC Information

Besides a general introduction to the IBM PC and IBM PS/2 in Chapter 1, Appendix D includes diskette formatting and operating instructions, and a list of popular magazines, newspapers, and manuals to help keep the student abreast of the many new developments in the personal computer field.

ANCILLARY MATERIALS

A comprehensive instructor's support package accompanies *Structured BASIC Fundamentals and Style for the IBM PC and Compatibles*. These ancillaries are available upon request from the publisher.

Instructor's Manual and Answer Book

The *Instructor's Manual and Answer Book* includes the following:

- Lecture outlines for each chapter
- Transparency masters from each chapter of the text
- Chapter-by-chapter objectives and vocabulary lists
- Answers to the odd-numbered Test Your BASIC Skills exercises
- Program solutions to the 60 programming assignments in the book
- Test bank, including true/false, short answer, fill-in, and multiple-choice questions for quizzes and tests

ProTest: An Easy-to-Use Computerized Test-Generating Package

Boyd & Fraser's state-of-the-art test-generating package, ProTest, has been designed specifically for this book. ProTest is an easy-to-use menu-driven package that is supplied on an IBM PC-compatible diskette. ProTest allows an instructor to create a customized test on the PC in a matter of minutes. The large test bank that accompanies ProTest includes field-tested true/false and multiple-choice questions. A user may also enter his or her own questions into the test bank.

ProTest will run on any IBM PC, IBM PS/2, or compatible system with two floppy-diskette drives or a hard disk.

Instructor Diskette

The Instructor Diskette includes the solutions to the 60 programming assignments found at the end of Chapters 2 through 10.

ACKNOWLEDGMENTS

We would like to thank and express our appreciation to the many fine and talented individuals who have contributed to the success of this book. We were fortunate to have a group of reviewers whose critical evaluations of our first four BASIC books, *Standard BASIC Programming, BASIC Fundamentals and Style, Complete BASIC for the Short Course,* and *Applesoft BASIC Fundamentals and Style,* were of great value during the preparation of these books. Special thanks again go to Professor James N. Haag, University of San Francisco; Professor R. Waldo Roth, Taylor University; Professor David Bradbard, Auburn University; Professor Donald L. Muench, St. John Fisher College; Professor Jerry Lameiro, Colorado State University; Professors John T. Gorgone, and I. Englander of Bentley College; Professor Chester Bogosta, Saint Leo College; Professor John J. Couture, San Diego City College; Professor Syed Shahabuddin, Central Michigan University; Sumit Sircar, University of Texas at Arlington; Marilyn Markowitz; and James Larson, director of computer services for the Homewood Flossmoor High School District in Homewood, Illinois. We are also very grateful to the following individuals, who reviewed the manuscript for *Structured BASIC Fundamentals and Style for the IBM PC and Compatibles:* Professor Al Schroeder, Richland College; Professor Dave Talsky, University of Wisconsin-Milwaukee; Professors George Fowler and Louise Darcey, Texas A&M University; Professor William Bailey, Casper College; Professor John Grillo, Bentley College; Professor Riki Kucheck, Orange Coast College; Professor Michael Walton, Miami-Dade Community College North;

Professor Dorelyn F. Anderson, Saint Cloud University; and Marjorie Leeson. The instructional staff of the Information Systems and Computer Programming Department of Purdue University Calumet provided many helpful comments and suggestions, and to them we extend our sincere thanks.

No book is possible without the motivation and support of an editorial staff. Therefore, our final acknowledgment and greatest appreciation are reserved for the following at Boyd & Fraser: Tom Walker, editor-in-chief, for the opportunity to write this book and for his constant encouragement; Donna Villanucci, who turned in an incredible developmental editing job; Becky Herrington, director of production, for her patience, diligence, and especially her creative talents; and finally, special praise for Toni Jean Rosenberg, for her invaluable editorial assistance.

Hammond, Indiana
January 1988

James S. Quasney
John Maniotes

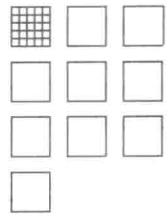
NOTES TO THE STUDENT

1. The first occurrence of a computer or programming term is printed in **boldface**. Its definition can be found nearby, usually in the same paragraph.
2. Beginning in chapter 4, line numbers appear near symbols in top-down charts and program flowcharts, in order to show their relationship to the corresponding program.
3. Every chapter ends with an important and useful review section called What You Should Know.
4. The answers to all the even-numbered Test Your BASIC Skills questions are at the back of the book, before the index.
5. All the executable programs in the text are on the Student Diskette that accompanies this book. The programs on the Student Diskette which correspond to those in the text begin with the prefix PRG, followed by the chapter and program numbers. For example, PRG2-8 refers to the eighth executable program in chapter 2.

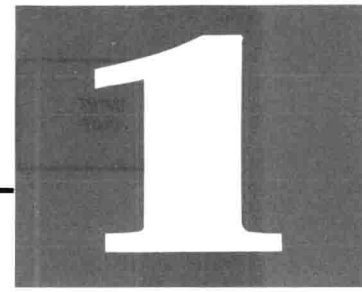
   Often, following the discussion of a program in the book, you will find a section labeled Try It Yourself. Follow the directions and load, modify, and execute the program. These short exercises will help you understand the significance of various Microsoft BASIC statements and how slight modifications to a program can affect the results.

   You will also find the programs on the Student Diskette helpful when you are solving assigned programming problems. These programs can be retrieved from the diskette, and statements can be added, modified, or deleted to arrive at a solution.
6. An easy-to-use reference card at the back of this book contains a summary of the Microsoft BASIC statements, commands, functions, special variables, special keys, operators, and reserved words.
7. Appendix A, "Program Flowcharting and Top-Down Design," and Appendix B, "Pseudocode and Other Logic Design Tools," provide you with additional valuable logic-design methods.
8. Appendix C, "Debugging Techniques, Programming Tips, and Chaining," suggests efficient methods for locating and correcting errors in a program. It also includes tips on how to write efficient, readable code and how to handle large programs.

TABLE OF CONTENTS



COMPUTERS AND PROBLEM SOLVING: AN INTRODUCTION



1.1 WHAT IS A COMPUTER?

A **computer** is a machine that can accept data, process the data at high speeds, and give the results of these processes in an acceptable form. A more formal definition of a computer is given by the American National Standards Institute (ANSI), which defines it as a device that can perform substantial computation, including numerous arithmetic and logic operations, without intervention by a human operator.

The key phrases in the ANSI definition are "substantial computation," "logic operations," and "without intervention by a human operator." These phrases point out the differences between computers and desk calculators.

The act of instructing a computer is known as **programming**. It involves writing precise instructions in a language the computer understands.

Advantages of a Computer

The major advantages of a computer are its speed and accuracy, as well as its ability to store and have ready for immediate recall vast amounts of data. Today's computers can also accept data from anywhere via telephone line or satellite communications. They can generate usable output, like reports, paychecks, and invoices, at a speed of several thousand lines per minute.

A recent statement made by James A. Allen of the public affairs division of Ford Motor Company illustrates the speed and accuracy of a computer. Mr. Allen's remarks concerned an engine control microcomputer system called Electronic Engine Control (EECIV), which the company says "can read seven engine parameters and change seven engine functions in less than one engine revolution — three-hundredths of a second. The computations performed during each minute of engine operation would take a human an estimated forty-five years or more using a manually operated calculator." This microcomputer system is made up of two chips, each less than a quarter of an inch square. In 1960, a computer with less capability than these two chips would have filled a room, and in 1970 it would have filled the trunk of a car.

Computers can handle tedious and time-consuming work and large amounts of data without ever tiring, which makes them indispensable for most businesses. In fact, computers have been among the most important forces in the modernization of business, industry, and society since World War II. Keep in mind, however, that with all their capabilities, computers are not built to think or reason. They extend our intellect, but they do not replace thinking.