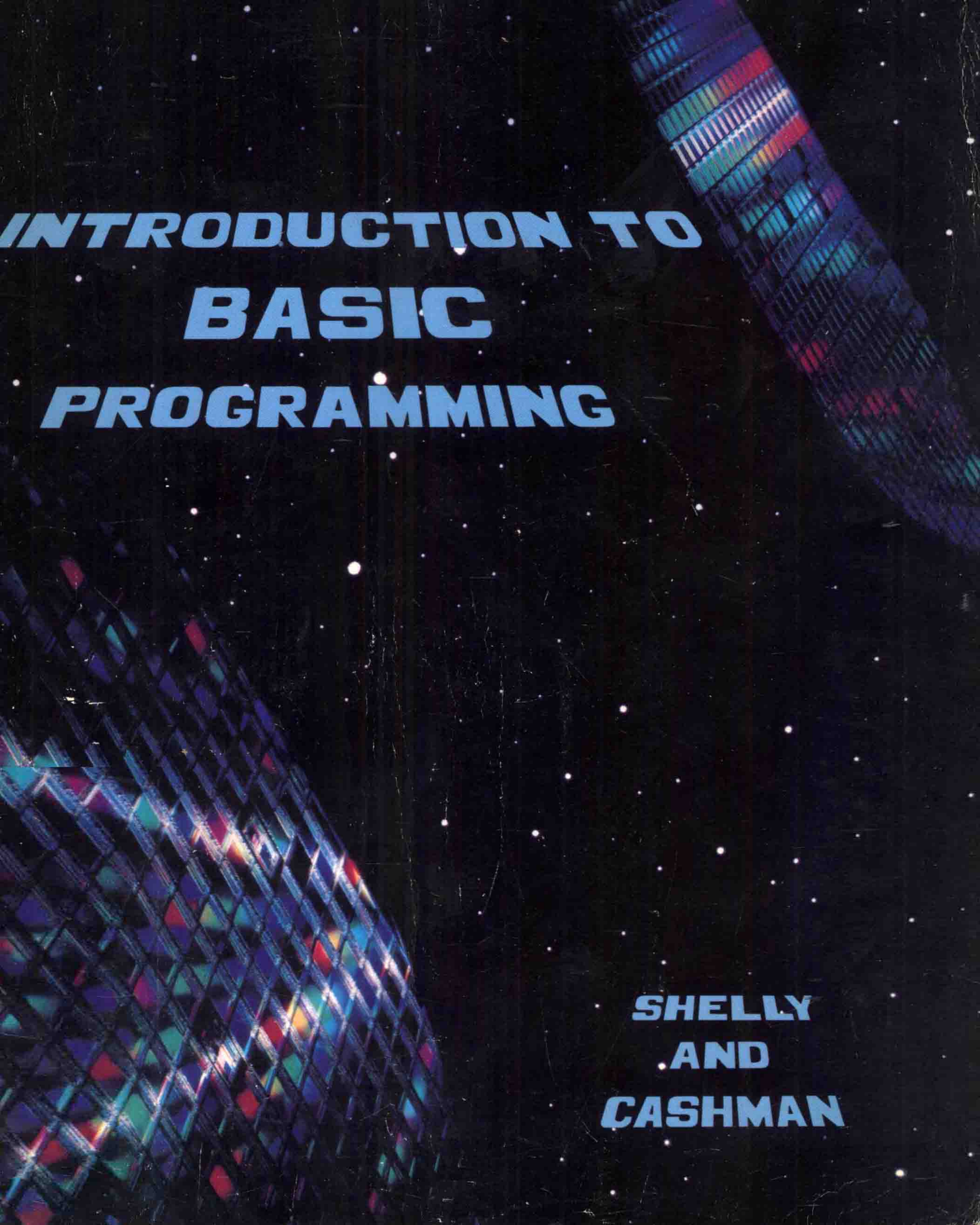
INTRODUCTION TO BASIC PROGRAMMING



SHELLY
AND
CASHMAN

INTRODUCTION TO BASIC PROGRAMMING

Gary B. Shelly
Educational Consultant
Brea, California

&

Thomas J. Cashman, CDP, B.A., M.A.
Long Beach City College
Long Beach, California

ANAHEIM PUBLISHING COMPANY
2632 Saturn St., Brea, CA 92621
(714) 993-3700

ANAHEIM
APC
PUBLISHING CO.

ISBN 0-88236-118-X

Printed in the United States of America

10987654321

INTRODUCTION TO BASIC PROGRAMMING

SHELLY & CASHMAN BOOKS

Introduction to BASIC Programming
Introduction to Computers and Data Processing
Introduction to Computer Programming Structured COBOL
Advanced Structured COBOL Program Design and File Processing
Business Systems Analysis and Design
Computer Programming RPG II
Introduction to Computer Programming ANSI COBOL
ANSI COBOL Workbook
Advanced ANSI COBOL Disk/Tape Programming Efficiencies
Introduction to Computer Programming RPG
Introduction to Flowcharting and Computer Programming Logic
Introduction to Computer Programming IBM System/360 Assembler Language
IBM System/360 Assembler Language Workbook
IBM System/360 Assembler Language Disk/Tape Advanced Concepts
DOS Utilities Sort/Merge Multiprogramming
OS Job Control Language
DOS Job Control for Assembler Language Programmers
DOS Job Control for COBOL Programmers
Introduction to Computer Programming System/360 PL/I

PREFACE

BASIC was developed in 1965 by Dr. John Kemeny and Dr. Thomas Kurtz at Dartmouth College for use in an academic environment. Its use has expanded considerably since then, particularly since its adoption in the late 1970's as the primary language on microcomputers. Today, it is probably the most widely used language in the world, with many minicomputers and over 1 million microcomputers using it as the main programming language.

When BASIC was developed in 1965, only a few of the characteristics of a good programming language as we know them today had been recognized. Further, little significant research had been published on how to design and write a good program. During that era, a program which produced the correct output was considered a good program. Little or no consideration was given to the need for reviewing a program at some future date.

In the years since BASIC was created, considerable maturity has occurred in the computing industry, particularly with respect to the programming process. The techniques of structured programming, with a heavy emphasis on developing logic using well-defined control structures and writing code which is easy to read, understand, and modify, have become widely accepted in the industry.

Unfortunately, this acceptance has not always been applied in the development of programs written in the BASIC language. Most periodicals serving those who use the BASIC language illustrate programs which are so poorly written that only the original programmer can decipher the processing taking place. Many BASIC textbooks illustrate trivial examples that violate the most fundamental rules of modern program design and coding. These books urge students to get on the computer and try different methods without instruction in the use of good program design and coding. Indeed, statements have even been made in textbooks that modern programming techniques cannot be taught when using the BASIC language. Such statements reflect a complete lack of understanding about the concepts of modern program design. Good program design is not dependent upon the programming language used. The programming language is merely the vehicle to implement the program design. Good program design using the control structures of structured programming can be implemented clearly and easily in any programming language.

The widespread use of BASIC as the first programming language taught to students and the use of poorly designed and written programs in the first programming course has led to considerable difficulty in later programming courses. For example, students who have been allowed or even encouraged to write programs in any format using any logic which will eventually work find the transition to a disciplined approach in programming quite difficult. It has been the authors' experience in talking with instructors that many students are unable to break the bad habits acquired in their first programming course using the BASIC language.

Clearly, then, there is a real need for a book which teaches good programming methodology using the BASIC language. This textbook is designed to teach not only the BASIC language but also the proper and correct ways to design and write programs.

The book uses a problem-oriented approach; that is, the student is introduced to computer programming through the use of a series of sample programs that are completely designed and coded. From the first program, students are shown the proper way to design and code a program using the BASIC language. Trivial examples that have no relation to an application are avoided. Heavy emphasis is placed on using only the three logic control structures found in structured programming: the sequence logic structure, the if-then-else logic structure, and the looping logic structure. By using these structures, the design and coding of the program is made much simpler than when a shotgun, haphazard approach is used.

Each sample program is thoroughly documented with remarks. All coding is indented following coding standards to improve the readability of the program listing. These concepts, although not new, are seldom found in textbook programs. They reflect a professional approach to the task of programming in the same manner that other educational disciplines require adherence to certain standards and rules. Indeed, it is difficult to imagine that an English teacher would allow a term paper to be submitted in any form desired by the student; for example, with no title page and no sentence or paragraph structure. Yet this is what has been occurring in BASIC programming classes for many years. The approach has often been . . . write the program in any way you want just to get it to work. By following the standards introduced in this text, students will emerge with a firm foundation in program design and will be able to produce professional quality software.

The book is designed to be used in a one quarter or one semester course in BASIC programming. No prior knowledge of data processing or computer programming is required. The first chapter provides a brief overview of basic computer concepts. An explanation is also included of the concept of a stored program and the process of developing a computer program. Each of the remaining chapters contains an explanation of an application, the program design and program code for the application, and a detailed explanation of the program design and the program code. In addition, each chapter contains an explanation of other BASIC statements and programming techniques related to the class of problem being studied.

There are many well-defined concepts which must be taught in an introductory programming class. These include basic input/output operations, basic arithmetic operations, accumulating and printing totals, comparing, array processing, searching and sorting, string processing, file processing, and report generation. In addition, modern computer systems are centered around transaction-oriented or interactive processing. In this type of environment, the user enters data directly into the computer system, processing occurs, and the results are immediately conveyed back to the user. Interactive processing many times includes the use of a menu and the editing of data entered from the keyboard. The sample programs in this textbook are unique because they illustrate all of these important concepts.

At the conclusion of each chapter, review questions, debugging exercises, and student programming assignments are provided. The review questions use a variety of methods to test the student's understanding of the material covered in the chapter. The debugging exercises present, for student review, both syntax errors and errors occurring in the execution of a program. The programming assignments should be designed and coded using the standards recommended in the textbook. These assignments range from applications very similar to those shown in the sample program of the text to more difficult programs which require creative thinking on the part of the programmer to arrive at a solution to the problem.

Upon completion of this textbook, the student will have gained experience and practice in designing and writing a variety of programs covering important programming techniques applicable to all disciplines. The emphasis on good program design and coding will lay the foundation for the subsequent study of other programming languages, as well as providing the student with a model of good software which may be used in analyzing and evaluating packaged software systems.

Acknowledgements

As in every book, there are key people without whom the effort could not be completed. For this book, we owe a sincere debt of gratitude to some very important people. The quality of color found in this book is a result of the skill of Quality Graphics in Santa Ana, California. Max Loftin and Sue Davis from Quality Graphics made a significant contribution to this book. The book is printed by R. R. Donnelley & Sons. We literally would not have this book were it not for the superb efforts of John Conley, Bob Rochelle, and Jane Kaitson from Donnelley.

Our primary gratitude must go to our two colleagues at Anaheim who on every book perform above and beyond the call of duty. Marilyn Martin, who typeset the book and performed an important role as color consultant, and Michael Broussard, who transformed our scribblings into masterful drawings and illustrations and laid out the book, are two people who continually make contributions that only the authors can truly understand and appreciate. This book is as much a part of them as it is of us.

Finally, it is appropriate to thank our spouses — Kathleen Shelly, Francis Cashman, Val Martin, and Marcia Broussard — for their patience and encouragement. Each of them has lived with this book longer than should be expected and for that we thank them.

Gary B. Shelly
Thomas J. Cashman

Picture Credits

The following people and organizations have contributed to the text. We thank them for their generosity.

Cover photograph: Photograph by Melgar Photographers, Santa Clara, CA for Advanced Micro Devices, Sunnyvale, CA
Cover lettering: Lisa Martin
Chapter 1 Title Page: International Business Machines Corporation

Figure 1–1	National Semiconductor Corporation
Figure 1–2	International Business Machines Corporation
Figure 1–3	Hewlett-Packard Company
Figure 1–4	Burroughs Corporation
Figure 1–6	Raytheon Company
Figure 1–7	Data Electronics, Incorporated
Figure 1–8	Cado Systems Corporation
Figure 1–9	3M Company
Figure 1–10	Comshare Incorporated

Title Pages

Chapter 2	Radio Shack, A Division of Tandy Corporation
Chapter 3	NCR Corporation
Chapter 4	National Semiconductor Corporation
Chapter 5	Digital Equipment Corporation
Chapter 6	Apple Computer Incorporated
Chapter 7	Perkin-Elmer Corporation
Chapter 8	National Semiconductor Corporation
Chapter 9	Honeywell, Incorporated
Chapter 10	Apple Computer Incorporated

CHAPTER ONE

COMPUTERS AND COMPUTER PROGRAMMING

OBJECTIVES:

Familiarization with computer systems

Familiarization with the basic data processing cycle — input, process, and output

Familiarization with the function of a computer program

An understanding of the operational capabilities of a computer system — input/output, arithmetic, and logical operations

A familiarization with the program development cycle

CHAPTER ONE

COMPUTERS AND COMPUTER PROGRAMMING

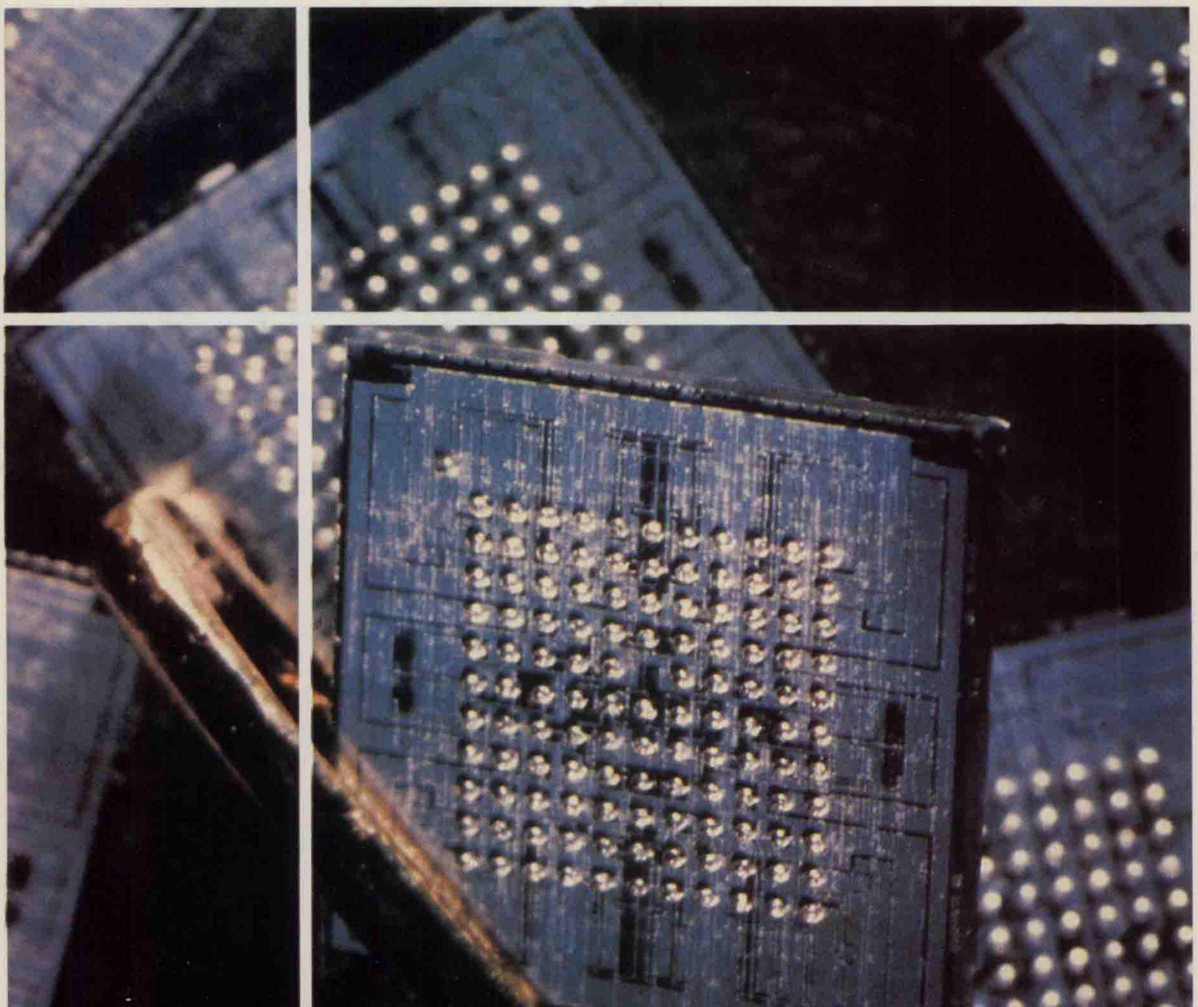


Table of Contents

CHAPTER ONE
COMPUTERS AND COMPUTER PROGRAMMING

CHAPTER TWO
INPUT / OUTPUT PROGRAMMING

CHAPTER THREE
ARITHMETIC OPERATIONS

CHAPTER FOUR
COMPARING

CHAPTER FIVE
MORE ON COMPARING

CHAPTER SIX
LOOPING — INTERACTIVE PROGRAMMING

CHAPTER SEVEN
ARRAYS

CHAPTER EIGHT
MENUS, ARRAYS, SUBROUTINES, AND SORTING

CHAPTER NINE
STRING PROCESSING

CHAPTER TEN
FILES, REPORT GENERATION, AND FUNCTIONS

Introduction

Today, the computer is an integral part of our daily lives. Airline reservations are made using a computer. A check used at the local department store may require computer verification. A bank teller is likely to use a computer terminal to find the balance in your account. Few aspects of our daily lives are left untouched by some type of computerized processing. With increasing frequency, children in elementary schools, students in high schools and colleges, and business executives everywhere are using the computer as a tool to assist in solving a variety of problems.

The ability to understand and use a computer system is rapidly becoming as important as the ability to read and write. People from all walks of life will routinely use computers in the future. The purpose of this book is to provide the ability to use many computer systems available today by teaching the BASIC computer programming language.

Types of computers

To most people, the computer is a mystery. This view may have arisen from the rapid manner in which the computer has entered our lives and the variety of computer systems that are found.

Computer systems of all sizes and capabilities are used today. Very small computer systems can be used to control electric appliances or regulate gas consumption in an automobile (Figure 1–1). Microcomputer systems are found in homes, businesses, and schools for a variety of uses (Figure 1–2). These machines provide new methods for learning, accessing information, and performing calculations.

Other computer systems, not much larger than an average desk, are found in thousands of businesses (Figure 1–3). They are used for business, scientific, and engineering applications.

Very large computer systems may be used to prepare thousands of telephone bills each day or to perform the millions of calculations required to make long-range weather forecasts. Such systems typically require large rooms housing what appears to be a bewildering array of electronic devices (Figure 1–4).

What is a computer

A computer is a device which can perform computations, including arithmetic and logic operations, without human intervention. To accomplish this, all computers perform basically the same functions regardless of their size. These functions are: