ROBERT C. NICKERSON

COBOL Programming

A STRUCTURED APPROACH

COBOL Programming

A Structured Approach

ROBERT C. NICKERSON
California State University, Hayward

WINTHROP PUBLISHERS, INC., Cambridge, Massachusetts

Library of Congress Cataloging in Publication Data

Nickerson, Robert C
COBOL programming.

Includes bibliographical references and index.
1. COBOL (Computer program language) 2. Structured programming. I. Title.
QA76.73.C25N5 001.6'424 76-58884
ISBN 0-87626-129-2

Cover design by David Ford

17 Dunster Street, Cambridge, Massachusetts 02138

10 9 8 7 6 5 4 3

Preface

This textbook is a carefully paced introduction to the COBOL programming language with an emphasis on good programming procedures. More specifically, the book has the following four main objectives:

1. To introduce basic concepts about computers, programming, and data.
2. To describe the COBOL programming language at a level of detail that allows the student to gain a working knowledge of the fundamentals of the language.
3. To explain and demonstrate the process of preparing programs with an emphasis on "structured" programming principles.
4. To explore programming techniques and computer applications.

The first objective is accomplished by providing introductory material on computers and data processing. No previous exposure to these topics is assumed. Chapters 1 and 2 present the essential background material necessary to begin studying COBOL.

The majority of the book is devoted to the second objective. The COBOL language is described carefully, and many examples are provided. The approach is to develop the concepts informally so that the student gains an intuitive feeling for the language. Sufficient material for coding simple programs is presented as early as possible in the book (Chapter 3). This material then is expanded in logical steps in subsequent chapters.

The version of COBOL that is emphasized is mainly a subset of the 1974 American National Standard (ANS) COBOL. The subset is compatible for the most part with the 1968 version of ANS COBOL. Where differences exist they are pointed out in the text and the reader is referred to Appendix A for further explanation. This appendix also summarizes implementation differences for several common computers. The machines discussed are the IBM System/360 and System/370, the IBM System/3, and the IBM 1130.

The third objective provides the most unique feature of the book. The process of developing a computer program that is understandable and correct is emphasized from the beginning. Sample programs with text discussion of the development process are used to demonstrate these ideas. The programming process is summarized in Chapter 8.

The approach to programming that is emphasized is commonly called "structured" programming. There is a lot of disagreement and misunderstanding about what structured programming means. In this book we use the definition that structured programming is a systematic process that results in programs that are easily understood, maintained, and modified, and that can be shown to be correct. We emphasize designing programs in a top-down manner. This approach leads to modular programs that are easy to understand and to change. In addition, top-down design helps us show the correctness of the program. We discuss program structure and style rules throughout the book and emphasize their importance in producing readable and correct programs.

COBOL is not the most ideally suited language for implementation of structured programming concepts. However, with some degree of care, it is possible to produce well-structured COBOL programs. One purpose of this book is to demonstrate the care that is required to accomplish this.

A problem that often occurs when attempting to use structured programming principles with a particular language is what to do with the GO TO statement. While some may feel that it is ideal to eliminate this statement completely from all programs, it is not always practical. The principle that is used in this book is to avoid using the GO TO statement unless the resulting structure is more complicated than it would be with a GO TO. For didactic reasons, the GO TO statement is introduced as the first control statement (Chapter 3). However, the PERFORM statement is presented shortly afterward (Chapter 5) to give the student an alternative and to allow better structuring of programs.

The final objective of the book is accomplished through examples and programming problems. Examples that demonstrate various programming techniques and applications are included throughout the text. Programming problems that require the use of these and other techniques are included in Appendix D. These problems range in difficulty from simple list-producing programs to complex disk, tape, and table-processing programs.

This book is oriented to business and data processing students at two-year and four-year colleges, and to students in business schools and specialized computer schools. In addition, practitioners in industry may find many of the structured programming topics of interest. The book assumes no mathematical or accounting prerequisites, and no previous exposure to computers and data processing is required.

The text may be used in a one- or two-semester (quarter) course. The amount of material that can be covered in a term depends to a large extent on the students' previous exposure to programming. If the course for which the book is used has no prerequisites, then eight to ten chapters can be covered in depth in one term. During the second term the remainder of the book can be covered, along with supplementary material from the references and from the manufacturer's manual. If the student audience has had previous exposure to another programming language, then most of the material in the book can be covered in one term.

The book is divided into three parts. Part One (Chapters 1 and 2) introduces the background material necessary to begin studying the details of COBOL. Part Two (Chapters 3 through 8) presents the fundamental elements of COBOL and develops the programming process. Part Three (Chapters 9 through 13) discusses advanced COBOL features. Parts One and Two are prerequisite to Part Three and should be read in sequence. The chapters in Part Three may be taken in any order after completing Part Two.

Four appendices are provided. Appendix A summarizes the important differences between 1968 ANS COBOL, 1974 ANS COBOL, and the versions of COBOL implemented on several popular computers. Appendix B is a list of COBOL reserved words. Appendix C provides instructions for the operation of the IBM 029 keypunch. Appendix D contains the programming problems.

Finally, there is a list of COBOL reference manuals and an annotated bibliography of readings dealing with the programming process and structured programming.

As is always true, the ideas in this book come from many sources. I cannot begin to thank the professors, writers, colleagues, and students who have contributed in some way to this book. The original idea for the book came from discussions with Mike Meehan, editor at Winthrop Publishers. During the book's development, Dennie Van Tassel gave me his unbiased opinion of many of my ideas. The final manuscript could not have been prepared without the diligent typing of Jay Munyer.

Perhaps the most significant contributions came from my wife, Betsy. She is decidedly my best typist, editor, critic, and friend.

Robert C. Nickerson
Aptos, California

Acknowledgment

The following information is reprinted from *COBOL Edition 1965*, published by the Conference on Data Systems Languages (CODASYL) and printed by the U.S. Government Printing Office.

Any organization interested in reproducing the COBOL report and specifications in whole or in part, using ideas taken from this report as the basis for an instruction manual or for any other purpose, is free to do so. However, all such organizations are requested to reproduce this section as part of the introduction to the document. Those using a short passage, as in a book review, are requested to mention "COBOL" in acknowledgment of the source, but need not quote this entire section.

COBOL is an industry language and is not the property of any company or group of companies, or of any organization or group of organizations.

No warranty, expressed or implied, is made by any contributor or by the COBOL Committee as to the accuracy and functioning of the programming system and language. Moreover, no responsibility is assumed by any contributor, or by the committee, in connection therewith.

Procedures have been established for the maintenance of COBOL. Inquiries concerning the procedures for proposing changes should be directed to the Executive Committee of the Conference on Data Systems Languages.

The authors and copyright holders of the copyrighted material used herein

FLOW-MATIC (Trademark of Sperry Rand Corporation), Programming for the Univac® I and II, Data Automation Systems copyrighted 1958, 1959, by Sperry Rand Corporation; IBM Commercial Translator Form No. F28-8013,

copyrighted 1959 by IBM; FACT, DSI 27A5260-2760, copyrighted 1960 by Minneapolis-Honeywell

have specifically authorized the use of this material in whole or in part in the COBOL specifications. Such authorization extends to the reproduction and use of COBOL specifications in programming manuals of similar publications.

Contents

PART ONE

Introductory Concepts

CHAPTER 1
Computers, Programming, and Data

To understand computer programming it is first important to be familiar with a few general ideas about computers, programming, and data. In this chapter, we discuss the basic background material that is necessary to begin studying programming. In subsequent chapters we will use this background information in our detailed discussions of computer programming.

1-1. BASIC CONCEPTS

Any calculating device can be called a "computer." For example, adding machines, slide rules, and desk calculators are all "computers" because each calculates or computes. However, the word "computer" usually is not used for such devices. Instead we restrict its use to a particular device that has three distinguishing characteristics.

First, computers are always electronic; that is, computers calculate by electrical means. The consequence of this characteristic is that computers can operate at very high speeds — electronic speeds.

The second important characteristic of computers is that they have the ability to retain facts and figures, called *data*, for future use; that is, the computer has a "memory," or, more correctly, an *internal storage* in which it can hold information. Internal storage

is not the same as human memory. However, like human memory, data can be placed in the computer's internal storage and then can be recalled at some time in the future.

Third, computers have the ability to retain in their internal storage a set of instructions that tells the computer what it is to do. Such a set of instructions is called a *program.* This program is prepared in advance by a person, called a *programmer*, who is familiar with the different things that the computer can do. The program that is prepared by the programmer is stored in the internal storage of the computer and is performed, or *executed*, automatically by the computer.

In this text we are concerned with the preparation of computer programs. The preparation process is called *programming*, and this book discusses a particular type of computer programming. However, before we can examine programming in detail, we must understand the physical organization of a computer.

1-2. ORGANIZATION OF A COMPUTER

Computers are distinguished from other devices by the characteristics described above. However, a computer requires a number of parts that have not been mentioned. For example, besides internal storage, there must be some unit in the computer that can interpret and execute instructions. In addition, there must be units to perform arithmetic, to make logical decisions, to put data into internal storage, and to retrieve data from storage.

We can view a computer as a system composed of several components. Figure 1-1 illustrates the important components of a computer and the relationship between them. Basically, a computer has three main parts — the input device, the central processing unit, and the output device.

An *input device* is a unit that accepts data from some source outside of the computer and transforms this data into electronic impulses. The data that is accepted is called *input data*, or simply *input.* One of the most common means of conveying input data to the computer is the punched card (see Figure 1-2). Data is recorded on the card by punching various patterns of holes into the card. (Details of punched card data are discussed later.) An input device for punched cards is designed to recognize what the patterns represent and to transform them into electronic impulses that are sent to the central processing unit. Such a device is called a *card reader* (see Figure 1-4).

An *output device* performs the opposite function of an input device. An output device transforms computer-created electronic

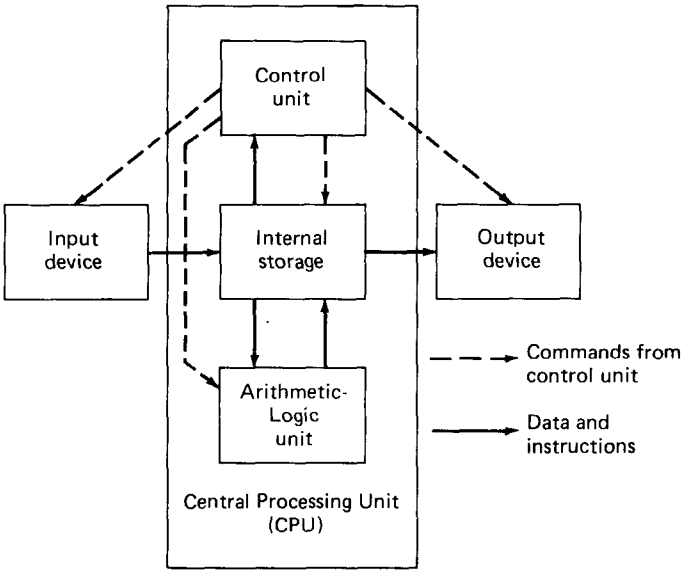

FIGURE 1-1. The Organization of a Computer

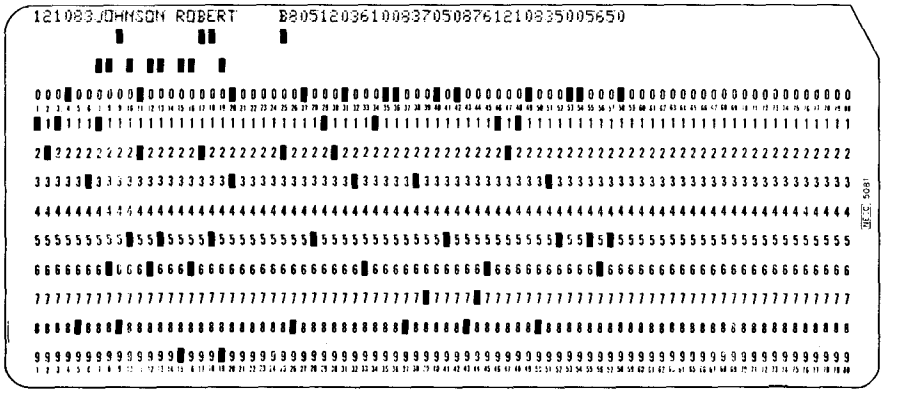

FIGURE 1-2. Punched Card Input

impulses from the central processing unit into a form that can be stored outside of the computer. The result is the *output data*, or simply the *output*, from the computer. A common form of output is a printed document or report (see Figure 1-3). Data from the central processing unit is transmitted by electronic impulses to a device called a *printer* (see Figure 1-4). In the printer these impulses are transformed into printed symbols, and a paper copy of the output data is produced.

Between the input device and the output device is the component that does the actual data processing and computing. This is the *central*

YEAR-TO-DATE SALES REPORT

SALESPERSON NUMBER	SALESPERSON NAME	SALES	RETURNS	NET
0005	BENNETT ROBERT	2,850.35	38.00	2,812.35
0016	LOCK ANDREW S	382.72	95.35	287.37
0080	PARKER JAMES E	30,700.14	555.00	30,145.14
0239	HAINES CYNTHIA L	101,000.00	2,200.00	98,800.00
0401	REDDING OLIVIA	156,159.15	24,052.64	132,106.51
0477	SMITH RICHARD A	1,450.00	510.00	940.00
0912	EMERY ELIZABETH G	36,200.35	1,730.15	34,470.20
1060	ROBINSON WILLIAM L	60,350.00	25.00	60,325.00
1083	JOHNSON ROBERT	63,311.96	893.55	62,418.41
1111	FREDERICKS RICHARD	52,600.00	483.50	52,116.50
1133	MARSHALL M S	210,000.00	.00	210,000.00
1205	HOLT BENTLEY	14,881.74	413.52	14,468.22
1374	BENTON ALEX J	2,600.13	267.50	2,332.63
1375	TAYLOR EVERETT	12,250.00	1,125.00	11,125.00
1420	EHRHARDT ELISE	4,890.64	981.00	3,909.64
1442	ADAMS JUNE R	96,771.46	1,572.36	95,199.10
1612	LOCATELLI FRANK	14,750.00	1,505.00	13,245.00
1698	GUZMAN JOSE	2,460.00	183.00	2,277.00
1842	COLE ROBERT N	306,650.39	81,637.92	225,012.47
	TOTALS	1,170,259.03	118,268.49	1,051,990.54

FIGURE 1-3. Printed Output