STRUCTURED COBOL PROGRAMMING

3/e

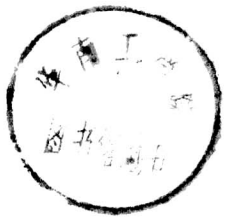
Stern/Stern

STRUCTURED COBOL PROGRAMMING

NANCY STERN
Hofstra University

ROBERT A. STERN
Nassau Community College

THIRD EDITION



JOHN WILEY & SONS

NEW YORK
CHICHESTER
BRISBANE
TORONTO

This book was set in Times Roman by Waldman Graphics.
It was printed and bound by Semline.
The designer was Fern Logan. The drawings were designed and executed by John Balbalis with the assistance of the Wiley Illustration Department. The copyeditor was Rosemary Wellner. Claire Egielski supervised production.

Library of Congress Cataloging in Publication Data:

Stern, Nancy B
Structured COBOL Programming
Second ed. published in 1975 under title: COBOL programming.

Includes index.
1. COBOL (Computer program language) 2. Structured programming. I. Stern, Robert A., joint author.
II. Title.

QA76.73.C25S75 1980 001.6'424 79-18434
ISBN 0-471-04913-1

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

STRUCTURED COBOL PROGRAMMING

TO LORI AND MELANIE

PREFACE

OBJECTIVES OF THE BOOK

This book has been written with three primary objectives. We wish to provide the beginner in data processing with (1) the ability to write efficient ANS COBOL programs using the structured approach, (2) an understanding of how COBOL is used effectively in commercial applications, and (3) the most efficient logical approach necessary for writing sophisticated programs. The approach used in this textbook differs from others on COBOL in the following ways.

1. This is neither a reference manual nor a programmed instruction textbook. Instead, it combines the advantages of both, while minimizing the disadvantages. This results in a combined text-workbook approach. Thorough explanations of each topic are contained here, with illustrations, questions, and answers immediately following. (For the exercises, the student should use a sheet of paper to cover up the answers. The asterisks indicate where the answers begin.) Several topics are presented with many illustrations and questions, so that the reader can relate the information and better understand the logical approach necessary for structured programming in ANS COBOL.

2. The organization of the text is most beneficial to the student. Most books on COBOL are fragmented, generally commencing with a discussion of the PROCEDURE DIVISION and leaving the rest for a later explanation. This makes it extremely difficult for beginners to understand how to organize a COBOL program effectively. They may understand each segment, but the relationship of one to another is difficult to conceptualize. To effectively utilize COBOL as a programming language, the programmer must understand this interrelationship. Therefore, with this book, the reader is able to write complete COBOL programs, however simple, after the first few lessons. Not only are segments of programs provided in the various explanations and illustrations, but previous learning is reinforced at all points. By giving complete illustrations and programs as answers to problems, the reader's conceptual understanding of COBOL is enhanced as is the ability to program in the language.

3. Illustrations, questions, and programs to be written by the reader are totally applicable to the commercial field. Most books in this field supply examples and questions that, although relevant to the particular points being explained, do not relate effectively to the business environment. As a result, the beginner does not fully understand the total applicability of ANS COBOL to business. We have overcome this problem by providing examples and programs that are realistic, in a business sense.

CHANGES IN THE NEW EDITION

STRUCTURED PROGRAMMING

The structured approach to COBOL programming has been introduced in a simple and straightforward manner. That is, instead of beginning with verbose justifications and explanations of why structuring is so popular and why it has been instituted to replace previous techniques, we simply present it as the programming method used in the book. In this way, students are given a thorough exposure to this method and come to recognize its intrinsic characteristics by themselves. Only in later chapters is structured programming given as an alternative to other techniques. Thus the student learns this efficient and effective technique from the start.

ADDITIONAL TOPICS

Several topics have been added to the book. These include:

1. Sequential file processing techniques, including control breaks, and tape updates.
2. The SORT feature.
3. The SEARCH statement as a method of table handling.
4. Validity Checking Routines.

Several topics have been expanded:

1. Disk processing
2. Appendixes
   a. Pseudocode as a method for representing logical control
   b. Job control

This added material must be understood if the student is to write intermediate-level as well as elementary-level programs. Since many COBOL courses are now two semesters these additions and expansions can be used for enhancing the student's appreciation and understanding of COBOL for these courses. The additional material is frequently useful for one-semester courses where advanced students are eager to go beyond the standard material presented.

COMPUTER PRINTOUTS

Computer printouts have been added for several reasons:

1. They make programs and program excerpts more readable.
2. They ensure the accuracy of illustrative material.
3. They familiarize the student with program listings and computer output.

FORMAT AND PEDAGOGIC APPROACH

There have been several reorganizations of sections within chapters, based on reviewers' comments and suggestions from our students. The COMPUTE section, for example, is now part of the arithmetic chapter.

The basic structure of the book, however, remains fundamentally the same. Summaries of each topic are provided as boxed units within each chapter. True-False review questions have been added to each chapter and some problems have been changed. In general, we firmly believe that the pedagogic approach presented here is well suited for college students and will provide a solid foundation for understanding the COBOL language.

ORGANIZATION OF THE BOOK

The book is now divided into six main units:

UNIT I: AN OVERVIEW OF COBOL

This unit introduces the student to structured programming in ANS COBOL. At the end of this first unit the student can write simplified COBOL programs. We believe the best way to learn the language is to write complete programs, however simple, from the beginning.

UNIT TWO: BASIC COBOL OPERATIONS

This unit contains a thorough discussion of the most frequently used COBOL verbs. It provides a thorough discussion of transfer of control and decision statements. After reading this unit, the student should be able to program elementary business applications with relative ease.

UNIT THREE: BASIC PRINT AND EDIT OPTIONS

Because printing reports is such an integral part of business applications and since COBOL provides so many options, an entire unit is devoted to this topic. After reading this unit, the student should be able to produce complex, management-level reports.

UNIT IV: ADVANCED LOGIC CONSIDERATIONS AND TABLE HANDLING ROUTINES

This unit includes a thorough discussion of all the methods that may be used to transfer control in a COBOL program. The rationale for using the structured approach is explained. Table handling, including the SEARCH statement, is presented in detail. This unit focuses on intermediate-level logic problems.

UNIT V: TAPE AND DISK PROCESSING

This unit concentrates on tape and disk processing. It has been significantly expanded from previous editions. Tape updates, merges, sorts, and control breaks are presented in detail. Disk processing is also emphasized. This unit is designed for the intermediate-level COBOL programmer.

UNIT VI: ADDITIONAL COBOL OPTIONS

This unit provides an overview of topics that are not necessary for intermediate-level COBOL programming but may facilitate such programming. It describes methods that may be used to simplify coding.

The Appendixes include standard COBOL entries such as reserved words, collating sequences, and formats. In addition, they provide a summary of three topics the student may or may not be familiar with: magnetic tape features, flowcharts and pseudocode, and job control. Students who have learned about these topics need not read these Appendixes; those with no previous exposure are encouraged to read them.

THE USE OF THE BOOK

This book is intended primarily for junior-college and four-year college students and requires no prior exposure to programming languages. We have provided no introduction to computer equipment because this equipment varies greatly among computer centers. In addition, COBOL is designed to be basically computer independent.

We express special appreciation to Burroughs, Honeywell, and IBM for their cooperation in supplying specifications, illustrations, examples, and photographs.

We would like to thank our editor, Gene Davenport, for his support, Ellen and Ilene Goldberg for their editorial assistance, Diane Zaremba for her help in preparing the instructor's manual, and Melanie and Lori Stern for preparing the index.

NANCY STERN
ROBERT A. STERN

ACKNOWLEDGMENT

The following acknowledgment has been reproduced from COBOL Edition, U.S. Department of Defense, at the request of the Conference on Data Systems Languages.

"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 compnay 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 (R) 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 or similar publications."

N.S.
R.A.S.

UNIT

1 AN OVERVIEW OF COBOL

CONTENTS

UNIT ONE: AN OVERVIEW OF COBOL

UNIT TWO: BASIC COBOL OPERATIONS

UNIT THREE: BASIC PRINT AND EDIT OPTIONS

UNIT FOUR: ADVANCED LOGIC CONSIDERATIONS AND TABLE HANDLING ROUTINES

UNIT FIVE: TAPE AND DISK PROCESSING

UNIT SIX: ADDITIONAL COBOL OPTIONS

CHAPTER 1

INTRODUCTION TO COBOL PROGRAMMING

A. COMPUTER PROGRAMMING

No matter how complex a computer may be, its actions are directed by individual computer instructions designed and tested by a computer **programmer.** The program consists of a set of instructions that will operate on input data and convert it to output. A computer, then, can operate only as efficiently and effectively as it is programmed.

All instructions to be operated on must be in machine language. For the programmer to code instructions in this form is very tedious and cumbersome. Memory addresses must be remembered, and complex numerical computer codes must be utilized.

Since programming in a machine language is so difficult, advances in programming technology were developed to enable the programmer to write Englishlike instructions. These instructions, however, must be translated or **compiled** into machine language before they can be executed. The computer itself performs this translation into machine language with the use of a control program.

Among the numerous **programming languages** that can be translated into machine form is COBOL, which is the one used most often for commercial applications.

The programmer, then, writes a set of instructions, called the **source program,** in one of the programming languages. It **cannot** be executed or operated on by the computer until it has been translated into machine language.

The source program is generally punched into cards by a keypunch machine. This **source deck** enters the computer and must be translated into a machine language program called the **object program** before execution can occur. A special program called a **compiler** translates source programs into object programs.

While the computer is performing this translation, any errors detected by the compiler will be listed. That is, any violation of a programming rule is denoted as an error. This type of error is sometimes referred to as a **syntax**

error. For example, if the instruction to add two numbers is spelled AD instead of ADD, the computer will print an error message. If errors are of considerable magnitude, translation will be terminated. Note that the errors detected during a compilation are **not** of a logical nature. A logic error is one in which the **sequence** of programming steps is not executed properly. The machine generally has no way of judging the logic in a program, but this may be tested by executing the program in a "trial run."

If errors are not present in the source program or only minor violations of rules occur, the translation process will continue until all instructions are in machine language form. The program can then be executed, or tested, at this point. If, however, execution is not desirable at this time, the object program may be saved by punching it into cards to obtain an object deck or by storing it on some other medium. Thus, this object program may be used to execute the instructions without the necessity to recompile. Figure 1.1 illustrates the steps involved in programming.

A program, therefore, specifies the logical sequence of computer instructions. When the logic of a program becomes complex, pictorial representations called **flowcharts** are drawn **prior to** the coding of the program. These pictorial representations illustrate program logic in a less complex manner, thus facilitating the writing of the program.

Such flowcharts will be illustrated throughout. For the beginner in data processing with no previous exposure to flowcharting, Appendix C provides an introduction to the basic concepts. This appendix also provides an introduction to **pseudocode,** which is another technique used to facilitate program coding.

SELF-EVALUATING QUIZ

Each question in the exercises will be followed by an asterisk which signals that the solution will follow. Use a sheet of paper to cover the solution when testing yourself.

1. The major task of a computer programmer is to _____.
   *****
   write and test computer instructions
2. A set of instructions that will operate on input data and convert it to output is called a _____.
   *****
   program
3. To be executed by the computer, all instructions must be in _____ language.
   *****
   machine
4. Programs are written in _____ language. Why?
   *****
   symbolic programming
   Machine language coding is cumbersome and tedious.
5. Programs written in a language other than machine language must be _____ before execution can occur.
   *****
   translated or compiled

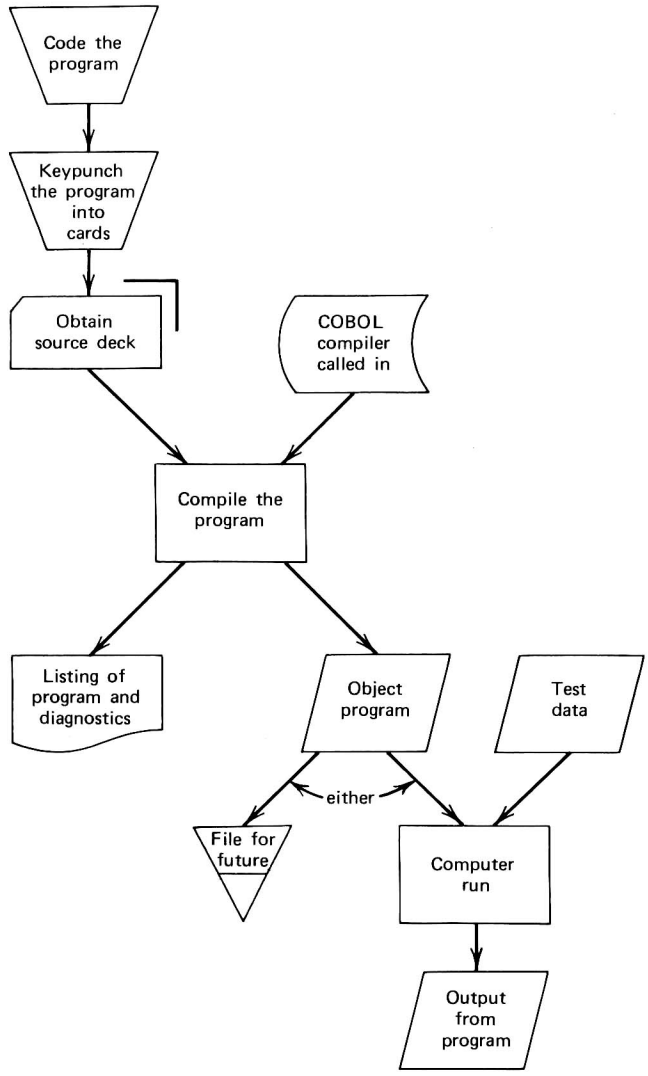


Figure 1.1 Steps involved in programming a computer.

6. _____ is an example of an Englishlike programming language.
   *****
   COBOL
7. _____ is the process of converting a COBOL program into machine language.
   *****
   Compilation
8. The program written in a language such as COBOL is called the _____ program.
   *****
   source
9. The source deck is the _____.
   *****