6th Edition

STRUCTURED COBOL PROGRAMMING

STERN & STERN

Free COBOL Syntax Reference Guide Included!

STRUCTURED COBOL PROGRAMMING

Sixth edition

Nancy Stern
Hofstra University

Robert A. Stern
Nassau Community College

John Wiley & Sons
New York Chichester Brisbane Toronto Singapore

Library of Congress Cataloging in Publication Data

Stern, Nancy B.
Structured COBOL programming / Nancy Stern, Robert A. Stern.—6th ed.
p. cm.
Includes bibliographical references.
ISBN 0-471-54929-0
1. COBOL (Computer program language) 2. Structured programming.
I. Stern, Robert A. II. Title.
QA76.73.C25S75 1991
005.13'3—dc20 90-40984
CIP

Printed and bound by Von Hoffmann Press, Inc.

10 9 8 7 6

STRUCTURED COBOL PROGRAMMING

To Lori Anne and Melanie

PREFACE

TO THE INSTRUCTOR

A. Overall Market

This book is intended for readers with no previous programming or computer experience as well as for those with some background in the computing field. It has been specifically designed for use in college courses on COBOL both in two-year and four-year schools.

B. Objectives of this Book

1. To teach students how to design programs so that they are easy to read, debug, modify, and maintain.
2. To provide students with the ability to write well-designed elementary, intermediate, and advanced structured COBOL programs in their entirety.
3. To familiarize students with information processing and systems concepts that will help them interact with users and systems analysts when designing programs.
4. To focus on the key elements of the most recent COBOL standard, called COBOL 85, that facilitate and promote the writing of well-designed structured programs. We highlight where COBOL 85 features differ from COBOL 74, the previous standard.
5. To familiarize students with programming tools such as pseudocode and hierarchy charts that make program logic more structured, modular, and top-down.

C. How This Book Differs from *Structured COBOL Programming*, Fifth Edition

The sixth edition of *Structured COBOL Programming* builds on the strengths of the previous five editions and includes some changes and additions we think have improved the text. The current dual emphasis on structured program design and syntax is retained, as is the focus on top-down modular programming and documentation. In this edition, we use structured pseudocode as well as standard pseudocode as a program planning tool.

The following are some of the specific changes we have made:

1. A manual is available entitled *Getting Started with RM/COBOL-85*. It focuses on Ryan McFarland (RM) COBOL-85 for IBM PCs and their

compatibles. We have written this manual in the belief that increasing numbers of users will want a microcomputer version of the language. An educational version of the RM/COBOL-85 compiler and text editor, along with key data and program files, will be supplied with this manual for students who wish to compile and run programs on an IBM or IBM-compatible microcomputer.

2. The text is streamlined so that it is more readable and less bulky, without adversely affecting the pedagogic approach or reducing the number of examples and illustrations used for reinforcing material presented.

   The overall appearance of the text has been improved so that it makes better use of space and color.

3. The quality of practice programs and programming assignments has been improved. Many practice programs and programming assignments have been changed so that they are more business-oriented, interesting, and challenging.

   Numerous programming assignments will use the same files for given application areas. This will help students understand the variety of programs required by specific types of organizations and will help tie together various aspects of business functions. We focus primarily, although not exclusively, on the Pass-Em State College student files, the We-Sell-Low Department Store, the Bon Voyage Travel Agency, The Video Trap: Movies for Less, Dollars and Sense Bank, and the International Cherry Machines Co. (ICM).

4. Many examples, illustrations, self-test questions, and review questions have been enhanced to make them more interesting and challenging. Many debugging exercises have been improved.

5. More complete data sets for programming assignments have been included.

   For many programming assignments we include data sets that have dozens of records, which is typical of most business applications. These will force page breaks and will include invalid data so that error-checking procedures will need to be included in programs.

6. The text will be both VAX- and IBM-oriented with discussions of VAX and IBM operating system commands and VAX and IBM text editors.

7. COBOL 85 has been fully integrated into the text. Key elements of COBOL 85 that make the language more suitable for structured programming are discussed in great detail. For example, the `EVALUATE` verb, scope terminators, the in-line `PERFORM`, and the methods used to avoid `GO TO`s are discussed in depth here. COBOL 85 is not simply presented as an addendum or supplement as in other texts. It has been fully integrated and highlighted with a shaded background to set it apart from COBOL 74.

8. Systems analysis and database design concepts such as data validation, error control procedures, file design, report design, and screen layout design are presented in detail. We believe students in programming courses should be familiar with these concepts. Knowledge of such systems concepts will provide a bridge to more advanced systems courses that students are apt to take later on.

9. The text will deemphasize or omit entirely all COBOL 74 and COBOL 85 features that have been put in the "obsolete" category, meaning they will no longer be supported in future standards. These include the `DATA RECORDS` clause, the `MOVE CORRESPONDING` statement, 77-level items, the `DEBUG` module, etc.

10. The text includes, in an appendix, a full update of all new features that have been approved by the ANSI Committee since 1985 for inclusion in COBOL 199x. Such material will help instructors in presenting new concepts.
11. A COBOL Syntax Reference Guide accompanies this text as a separate booklet. This guide can be used by students for "hands-on" sessions at terminals and PCs. It focuses on ANSI COBOL 85 but also includes VAX and IBM enhancements.
12. Standard pseudocode continues to be illustrated along with COBOL 85's in-line `PERFORM`. Structured pseudocode is also emphasized as a program planning tool.
13. Program coding required for printing well-designed reports is explained in detail as early as Chapter 6. In this way, students learn appropriate coding rules for printing output before they develop any bad habits.
14. Using a spiral approach, we sketch out basic instructions beginning in Chapter 4 so that students can write meaningful programs early on. Then we build on these instructions as we proceed through the text.
15. The inside front cover includes a quick reference to interpreting instruction formats.

D. The Pedagogic Approach

This book basically follows CIS-2 and CIS-3 of the DPMA Model Curriculum and COMP.4 of the Associate-Level Model Curriculum.

In this, as in our other programming books, we have attempted to write a stand-alone text with all the instructional material, examples, self-tests, and exercises the student needs to learn the language.

We have provided students with a step-by-step introduction to the subject, one that has thorough explanations followed by programs or program excerpts that reinforce and illustrate all concepts.

Each concept is thoroughly explained and illustrated with actual examples. Self-test questions, designed to help students evaluate their understanding of the material, appear after major concepts are discussed and also at the end of each chapter.

The book is segmented into *units*, each of which has a specific focus. After completing the first unit, students will be able to write elementary COBOL programs *in their entirety*. Subsequent units focus on program design and build up to more advanced programming concepts.

E. Instructional Aids

An instructor's resource manual is available. It contains (1) solutions to all text questions and programming assignments, (2) full examinations, and (3) course outlines. It is packaged with a disk that contains (1) test data for all programming assignments and (2) additional debugging exercises. Transparency masters are also available. A computerized test bank is available as well.

The educational version of Ryan McFarland's (RM) COBOL-85 is packaged with our manual called *Getting Started with RM/COBOL-85.*

The reviewers who provided many helpful suggestions throughout the development of this project are acknowledged on page xiii. We also thank Carol L. Eisen for her invaluable assistance in the preparation of the manuscript. Our special thanks to (1) the following individuals at John Wiley and Sons:

Joseph Dougherty, Editor, and Suzanne Ingrao, Director of Production; (2) the following individuals at Hudson River Studio: Ed Burke, Design, and Bob Crimi, Project Manager; (3) Shelley Flannery and Betty Pessagno, Copy Editors and Proofreaders; (4) Eric Carmeli, Technical Proofreader.

One last word of thanks to Hofstra University for giving us the opportunity to experiment with some new ideas and techniques, and to our students, whose interesting and insightful questions helped us improve our pedagogic approach.

We update our programming texts every few years and welcome your comments, criticisms, and suggestions. We can be reached c/o:

Nancy Stern
Robert A. Stern
BCIS Department
Hofstra University
Hempstead, NY 11550

You can also contact us using CompuServ's EasyPlex electronic mail service. Our user id is 76505,1222.

Our Bitnet address is ACSNNS@Hofstra.

PREFACE

TO THE STUDENT

Goals

The primary goal of this book is to teach you how to design and write COBOL programs. To accomplish this, we focus on two topics: (1) how programs are best structured and organized, and (2) the rules for programming in COBOL. Learning how to design programs will provide you with the basic tools for writing a program in *any* language. Learning the COBOL rules will specifically prepare you for writing programs in the COBOL language.

Features of the Text

Format

The format of this text is designed to be as helpful as possible. Each chapter begins with:

1. **A detailed chapter outline.**
   Before beginning a chapter, you can get an overview of its contents by looking at this outline. In addition, after you have read the chapter, you can use the outline as a summary of the overall organization.
2. **A list of objectives.**
   This list helps you see what the chapter is intended to teach even before you read it.

The material is presented in a step-by-step manner with numerous examples and illustrations. Within each chapter there are self-tests, with solutions, that are designed to help you evaluate your own understanding of the material presented. We encourage you to take these tests as you go along. They will help pinpoint and resolve any misunderstandings you may have.

End-of-Chapter Material

Each chapter ends with learning aids consisting of:

1. *Chapter Summary.*
2. *Chapter Self-Test*—with solutions so you can test yourself on your understanding of the chapter as a whole.
3. *Practice Program*—a full program is illustrated. We recommend you read the definition of the problem and try to code the program yourself. Then compare your solution to the one illustrated.

4. *Key Terms List.* This is a list of all new terms defined in the chapter. Appendix H is a glossary that lists all key terms in the text along with their definitions.
5. *Review Questions.* These are general questions that may be assigned by your instructor for homework.
6. *Debugging Exercises.* These are program excerpts with errors in them. You are asked to correct the coding. The errors highlighted are those commonly made by students and entry-level programmers.
7. *Programming Assignments.* The assignments appear in increasing order of difficulty. They include a full set of specifications similar to those that programmers are actually given in the "real world." You are asked to code and debug each program using test data. You will need to either create your own test data or receive a set from your instructor. Appendix C includes sample test data that can be used for Programming Assignment 2 in each chapter.

A manual entitled *Getting Started with RM/COBOL-85* is available with this text, along with an educational version of the RM/COBOL-85 compiler for IBM and IBM-compatible microcomputers. It focuses on Ryan McFarland (RM) COBOL. See your instructor for information on how to obtain the manual and disks, or contact Joseph Dougherty, Editor, John Wiley and Sons, 605 Third Avenue, New York, NY 10158.

Assumptions about the Reader

This book has been written on a level that is appropriate for introductory computer students. No previous programming experience is required. For those with no background, we encourage you to read the text in sequence and pay particular attention to end-of-chapter tools for reinforcing and testing your knowledge.

If you know another programming language, you may be able to proceed more quickly through the text, but we do not believe you will find it simplistic or, worse, boring. Because we emphasize program design features and aim to teach you not only syntax but proper programming form, we hope you will find the approach more conceptual than you might have experienced with other texts. Accordingly, we recommend that you read the text in sequence and skim the end-of-chapter material.

We update our programming texts every few years and welcome your comments, criticisms, and suggestions. We can be reached c/o:

Nancy Stern
Robert A. Stern
BCIS Department
Hofstra University
Hempstead, NY 11550

Acknowledgments

We thank the following reviewers for their many helpful suggestions: David E. Douglas, University of Arkansas; Carol C. Grimm, Palm Beach Community College; James P. Ley, University of Wisconsin, Stout; Barbara J. Maccarone, North Shore Community College; Barry Martin, Southeast Community College; Dan Rota, Robert Morris College; and Bob Spear, University of Maryland, College Park. We would also like to thank the following individuals for their contributions during the developmental stage of this project: Roger L. Anderson, College of Lake County; Madeline Baugher, Southwestern Oklahoma State University; Pierre Bettelli, Mesa State College; Helen Casey, Sam Houston State University; Don Disler, Belleville Area College; Frances Grodzinski, Sacred Heart University; Robert Hogue, Youngstown State University; Marilyn Meyer, Fresno City College; Margaret Porciello, SUNY College at Farmingdale; Cindy Pryke, Commonwealth College; Janet Urlaub, Sinclair Community College.

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 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 (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.

CONTENTS

Notes:
1. Full programs will be illustrated and assigned beginning with Chapter 1.
2. A debugging section with practice problems is in each chapter beginning with Chapter 2.

UNIT I THE BASICS

1. An Introduction to Structured Program Design in COBOL 2
2. COBOL Language Fundamentals: The IDENTIFICATION and ENVIRONMENT DIVISIONs 32
3. The DATA DIVISION: FILE SECTION and WORKING-STORAGE SECTION 54
4. Designing Complete COBOL Programs: A Closer Look at the PROCEDURE DIVISION and the PERFORM Statement 96

UNIT II DESIGNING STRUCTURED PROGRAMS

5. The Theory of Structured Program Design 124
6. Moving Data and Printing Information 171
7. Computing in COBOL: The Arithmetic Verbs 226
8. Selection Using the IF Statement 260
9. Iteration: Beyond the Basic PERFORM 298

UNIT III WRITING HIGH-LEVEL COBOL PROGRAMS

10. Control Break Processing 330
11. Debugging Programs and Validating Data 378

UNIT IV ARRAY PROCESSING AND TABLE HANDLING

12. Single-Level Arrays and Tables 412
13. Multiple-Level Arrays and Tables 462

UNIT V FILE MAINTENANCE

14. Sequential File Processing 494
15. Sorting and Merging 533
16. Indexed File Processing 567
17. Interactive Processing 606

UNIT VI ADVANCED TOPICS

18. Using Advanced Debugging Aids and Improving Program Performance 634
19. The COPY and CALL Statements 644
20. The Report Writer Module 661
21. Relative File Processing 688

APPENDIXES
A. COBOL Character Set and Reserved Words 706
B. VAX and IBM Operating Systems and Text Editors 710
C. Data Set for Programming Assignment 2 in Each Chapter 717
D. Major Differences between COBOL 85, COBOL 74, and the Proposed New COBOL Standard (199x) 718
E. Gateways to Other Programming Concepts: Database Management Systems and Object-Oriented Programming 723
F. A Brief Overview of CICS 734
G. An Overview of Microsoft COBOL and RM/COBOL-85 for the IBM PC 737
H. Glossary 742
A Review of COBOL 85 and COBOL 74 Features as Highlighted in this Book 753

Index 755
Interpreting Instruction Formats (inside front cover)
Program Design Features (inside back cover)

Accompanying this text is a COBOL Syntax Reference Guide, coding sheets, and Printer Spacing Charts.

This text is available with a student edition of the RM/COBOL-85 compiler version 4.0, which is Ryan McFarland's COBOL compiler for PCs, and a manual entitled *Getting Started with RM/COBOL-85.*

UNIT 1

THE BASICS

An Introduction to Structured Program Design in COBOL

I. COMPUTER PROGRAMMING: AN OVERVIEW
  A. Types of Computer Programs
  B. Applications Programs
  C. Machine Language Programs
  D. Symbolic Programs
II. THE APPLICATIONS PROGRAM DEVELOPMENT PROCESS
  A. An Overview
  B. Obtaining Program Specifications
  C. Using Program Planning Tools
  D. Coding the Program
  E. Compiling the Source Program
  F. Testing the Program
    1. Debugging Phases
    2. Debugging Techniques
III. THE NATURE OF COBOL
  A. COBOL as a Business-Oriented Language
  B. COBOL as a Standard Language
  C. COBOL as an English-like Language
  D. COBOL as a User-Friendly Language
IV. A HISTORY OF COBOL AND THE ANS VERSIONS
  A. When It Began
  B. The American National Standards (ANS) Versions of COBOL
  Self-Test
V. TECHNIQUES FOR IMPROVING PROGRAM DESIGN
  A. Structured Programming Using Modular Design for Coding Paragraphs
  B. The Top-Down Approach for Coding Modules
VI. A SAMPLE PROGRAM
  A. An Overview of the Four Divisions
  B. Definition of the Problem
  C. Input Layout
  D. Output Layout
  E. The Program Illustrated
    1. Reviewing the Specifications
    2. Coding Rules
    3. The IDENTIFICATION and ENVIRONMENT DIVISIONs
    4. The DATA DIVISION
    5. The PROCEDURE DIVISION
  F. A Brief Overview of Program Planning Tools
VII. Keying and Running a COBOL Program on Your Computer
END-OF-CHAPTER AIDS
  Chapter Summary
  Chapter Self-Test
  Key Terms
  Review Questions
  Programming Assignments