

ELEMENTS OF COBOL PROGRAMMING

WILSON T. PRICE
JACK L. OLSON

ELEMENTS OF COBOL PROGRAMMING

TP31
P3

7860388

Elements of cobol programming/

TP31
P3

7860388

**WILSON T. PRICE
JACK L. OLSON**

ELEMENTS OF COBOL PROGRAMMING



E7860388



THE DRYDEN PRESS
HINSDALE, ILLINOIS

ACKNOWLEDGMENT

The following extract from Government Printing Office Form Number 1965-0795689 is presented for the information and guidance of the user:

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 or similar publications.

Copyright © 1977 by The Dryden Press,
a division of Holt, Rinehart and Winston, Publisher.

All rights reserved

Library of Congress Catalog Card Number 77-187112

ISBN: 0-03-018371-5

Printed in the United States of America

7 8 9 140 9 8 7 6 5 4 3 2

*To our patient wives
Jean and Irene*

CONTENTS

PREFACE 11

INTRODUCTION

A FEW WORDS TO THE READER

- A Story of Success 17
- Horse Racing and Data Processing 18
- Writing Computer Programs 20
- Programming Languages 22

CHAPTER 1

BASIC PRINCIPLES OF COBOL

- A Typical Data Processing Problem 23
 - Problem Definition 23
 - Problem Characteristics 24
 - A Problem Solution—Example 1-1 26
- The DATA DIVISION 26**
 - The File Description (FD) 27
 - Level Numbers 27
 - The Input Record Description 27
 - The Output Record Description 29
 - Words and Names in Cobol 29**
 - Reserved Words 30
 - Programmer-Selected Names 30
 - The Identification and Environment Divisions 32**
 - Identification Division 32
 - Environment Division 32
 - The Procedure Division 33**
 - Cobol Commands 33
 - The MOVE Statement 34
 - Initializing the Output Line 36
 - Preparing Data Files For Use 36
 - The READ Statement 37
 - The WRITE Statement 38

- Transfer of Control and Paragraph-Names 38
- The Cobol Coding Form (and Other Details) 39**
 - The Coding Form 39
 - A- and B-Area Entries 41
 - Formatting and the Use of Punctuation 42
- Flowcharting 43**
 - Flow Direction 44
 - Input/Output 44
 - Processing 44
 - Termination 45
 - Flowcharting 45
- Answers to Preceding Exercises 46
- Programming Problems 47

CHAPTER 2

ARITHMETIC OPERATIONS

- Updating an Inventory 50
 - Problem Definition 50
 - A Program—Example 2-1 51
- Data Division Entries 52**
 - The Numeric Data Descriptor 52
 - The Edit Symbol 53
- The WORKING-STORAGE SECTION 54**
 - Independent Items 54
 - The VALUE Clause 55
- Arithmetic Operations 56**
 - Addition 56
 - Subtraction 58
 - Summary of Add and Subtract 59
- Data Involving Decimal Points 60**
 - Problem Definition 60
 - A Program—Example 2-2 61
 - Decimal Point Positioning on Input 61
 - Editing Output Quantities 62
- Multiplication and Division 64**

The MULTIPLY Statement 64
The DIVIDE Statement 66
Answers to Preceding Exercises 68
Programming Problems 69

CHAPTER 3

PREPARATION OF REPORTS

Using Records from Working-Storage Section 72
Characteristics of the Printer Record 72
Data Movement 73
Printing a Simple Report 74
Output Format Planning 75
A Program 76
Improving the Program 77
The WRITE-FROM Option 77
Defining Literals 78
Continuation of Literals 78
Printed Output—Page Positioning 79
The ADVANCING Option 79
Positioning To a New Page 81
Special-Names 82
Answers to Preceding Exercises 83
Programming Problems 84

CHAPTER 4

DECISION MAKING

Simple Program Loops 87
Controlling a Loop 87
Illustrating Decisions with Flowcharts 88
Basics of the IF Statement 91
Some Program Examples 91
Condition Tests 93
More on the IF Statement 94
A Basic Form of the IF 94
The ELSE Option 95
Programming Examples Using the IF 97
Printer Page Control 97
Program Planning 98
A Program—Example 4-4 99
A Program Flowchart and Program—Example 4-5 100
A Program Flowchart and Program—Example 4-6 102
Answers to Preceding Exercises 103
Programming Problems 104

CHAPTER 5

THE PROGRAMMING CYCLE

The Modern Digital Computer 107
Computer Hardware 107
Operating Systems 108
Computer Languages 110

The Process of Compiling 111
A Problem of Communication 111
Error Detection 113
A Philosophy Regarding Errors 115
Program Testing 115
General Considerations 115
Test Data 116
Answers to Preceding Exercises 118

CHAPTER 6

EXTENDING THE BASIC PRINCIPLES

The Cobol Language Structure 119
Standard Cobol Statement Form 119
Program Structure 121
Identification Division 122
The Environment Division and I/O Operations 123
General Structure 123
Input/Output Operations 125
Data Division 129
The FD Entry 129
The Record Description 130
Qualification of Data-Names 130
Move Corresponding 131
The REDEFINES Clause 133
The Procedure Division 136
Characteristics of the MOVE 136
Comparing Fields 138
Remarks on Documentation 139
Answers to Preceding Exercises 140

CHAPTER 7

MORE ON CONDITIONAL BRANCHING

PART 1 ADVANCED FEATURES OF THE IF 142

Basic Concepts of Loops 142
Simple Loops 142
Variable Loop Control 145
Loop With Multiple Exits 147
Further Capabilities of the IF 147
Review of Basic Principles 147
The Sign Test and the Class Test 149
Compound Conditional Statements 151
Examples of Compound Conditional Statements 153
Satisfying Either of Two Conditions 153
Satisfying Both Conditions of Two 154
Hierarchy of Operations 154
Implied Subjects and Operators 156
The Implied Subject 156
The Implied Operator 158
Other Concepts 160
Nested IF Statement 160

Condition Names 162

PART 2 APPLICATIONS AND ADVANCED CONCEPTS 165

Programmed Switches 165

The Concept of Programmed Switches 165

Examples of Programmed Switches 167

Commentary on the Program Switch 170

Minor Totals 170

Control Groups 170

Minor Total Logic 172

The PERFORM Statement 176

Answers to Preceding Exercises 180

Programming Problems 183

CHAPTER 8

ADVANCED ARITHMETIC OPERATIONS

Data Representation 188

The Basic Unit of Storage 188

Data Formats 190

The USAGE Clause 192

The SYNCHRONIZED Clause 194

Representation of Negative Numbers 195

Arithmetic Operations within the Computer 196

Decimal Alignment 196

Truncation and Rounding 198

Detection of Arithmetic Overflow 201

Summary 203

The COMPUTE Statement 203

Introduction to the COMPUTE Statement 203

Evaluation of Arithmetic Expressions 204

Rounding, Size Errors, and Intermediate Results 206

Summary 207

Answers to Preceding Exercises 207

Programming Problems 209

CHAPTER 9

ADVANCED REPORT FUNCTIONS

Page Control 213

The Concept of Page Control 213

Carriage Control 215

The LINAGE Clause 219

Toward Producing More Readable Reports 224

Report Headings 224

The ACCEPT and DISPLAY Statements 226

Column Headings 228

Editing 229

How the PICTURE Works 229

Data Types 232

Summary of Editing 234

Other Techniques 235

Variable Line Spacing 235

IBM Carriage Control 238

A Final Thought 238

Answers to Preceding Exercises 240

Programming Problems 241

CHAPTER 10

STATEMENTS WITH BUILT-IN LOOPS

The PERFORM Statement 243

Multiparagraph Procedures and the EXIT 243

The UNTIL Option of the PERFORM 246

The TIMES Option of the PERFORM 247

The VARYING Option of the PERFORM 249

EXAMINE, INSPECT, and TRANSFORM 250

The EXAMINE Statement 250

The INSPECT Statement 251

The TRANSFORM Statement 253

Answers to Preceding Exercises 254

Programming Problems 255

CHAPTER 11

SUBSCRIPTING AND TABLES

Basic Principles of Tables 261

The Notion of Subscripting 261

The Use of Subscripts in Cobol 261

The OCCURS Clause 264

Practical Aspects of Loading a Table of Data 265

Using the PERFORM 268

Predetermined Tables in Storage 268

Table Searching 269

The Concept of Arguments and Functions 271

Finding Pay Rate, Given Job Code 272

Multiple-Level Tables 275

The Cobol Table-Handling Feature 277

A Program Using the Search Feature 277

The OCCURS DEPENDING ON Clause 277

The SET Statement 280

The SEARCH Statement 280

The SEARCH ALL Statement 281

Answers to Preceding Exercises 283

Programming Problems 285

The Concept of a Record Key 294

CHAPTER 12

DISK AND TAPE OPERATIONS

General Considerations 289

Sequential and Random Processing 289

Magnetic Tape 290

Magnetic Disk 292

File Organizations 294

- Sequential File Organization 295
- Direct File Organization 296
- Indexed File Organization 298
- Relative File Organization 300
- Summary of File Organization Methods 301
- Blocked Records 302**
- Coding for Disk and Tape Operations 304**
 - The SELECT Clause 304
 - The File Description 305
 - The OPEN Statement 306
 - The READ Statement 307
 - The WRITE Statement 308
- Other Magnetic Disk Instructions 308
- Example Programs 309**
 - Processing an Indexed File 309
 - Processing a Direct File 312
- Summary 315**
- Answers to Preceding Exercises 315**
- Programming Problems 316**

CHAPTER 13

THE COBOL SORT FEATURE

- The Concept of Sorting 320**
 - Introduction 320
 - Sorting by Computer 322
- Cobol Sort Feature 325**
 - Coding the Sort Feature 325
 - The Identification Division 327

- The Environment Division 327**
 - The Data Division 327
 - Procedure Division Coding 328
- Sort Keys and Sort Sequence 329**
 - Sort Sequence 329
 - The Collating Sequence 329
 - Sort Keys 331
- Using Procedures with the Sort 333**
 - Input Procedures 333
 - Output Procedures 336
- Answers to Preceding Exercises 339**
- Programming Problems 341**

CHAPTER 14

MODERN PROGRAMMING TECHNIQUES

- Introduction 343**
- Top-Down Program Design 344**
- Structured Programming 348**
 - Summary 356**
- Suggested Readings 357**
- Answers to Preceding Exercises 357**

APPENDIX 1

COBOL RESERVED WORDS 359

APPENDIX 2

COBOL STATEMENTS—GENERAL FORMAT 363

INDEX 371

PREFACE

This book, as its title suggests, presents the elements of the computer programming language called Cobol. Thus, it seems safe to assume that the reader is interested either in learning the Cobol language or in teaching it. But before either of these activities is begun, the reader should be aware of the answers to several questions:

1. What is Cobol?
2. Why should I bother to learn (or teach) it?
3. How will this book help me to learn (or teach) the Cobol language?

Cobol is a programming language that was developed in the late 1950s and early 1960s in response to an urgent need. Computers, although less than ten years old, had taken firm hold in the data processing industry. Businesses and government agencies were heavily committed to "EDP" (or electronic data processing, as computerized data processing was then called) and were finding, to their consternation, that the computer world was a veritable tower of Babel: there were more programming languages than there were computer manufacturers. Furthermore, the programs used to communicate with one manufacturer's computers could not be understood by those of another manufacturer. To remedy this situation, the CODASYL (Conference on Data Systems Languages) committee was formed and charged with the development of a programming language which could be used with any computer. The result was the *COmmon Business-Oriented Language*, better known as *COBOL*.

Cobol, as specified by the CODASYL committee, was far from ideal: it had several drawbacks and a certain clumsiness, and it never achieved the complete machine independence desired by its designers. But it passed the two most important tests: it worked, and it was accepted by the business and government communities. And this is the answer to the second question; Cobol is worth learning because it is by far the most widely-used programming language for business applications. Any person who masters a programming language has a marketable skill since the market is far wider for Cobol than for any other language.

We can assume, then, that the reader is convinced of the desirability of learning Cobol and is eager to commence. But how should one begin? What is the best way to learn the Cobol language? The answer is simple: by writing programs in that language. After all, learning a programming language is no different from learning any unknown language: mastery comes most quickly if the language is used as soon as the basic grammar and vocabulary have been learned. For that reason, this book introduces the student to a

complete Cobol program containing all of the necessary elements for input and output in Chapter 1. Upon completion of Chapter 1, the student is ready to write similar complete programs which can be run on a computer to produce meaningful results. Subsequent chapters expand upon the groundwork laid in Chapter 1 to present further features and refinements of the language.

There is a saying which goes something like this: "I hear, and I forget; I see, and I remember; I do, and I understand." This book implements that philosophy by giving the student plenty of opportunities to see and to do. Examples are used liberally to illustrate the theoretical material presented in the text. Whenever possible these examples are not mere fragments of coding but are complete programs with input and output format specifications, flowcharts, and coding, so that the student can see the interrelationship of all of the components of the program.

To give the student the opportunity to learn by doing, two means are employed: in-chapter exercises and end-of-chapter problems. The in-chapter exercises appear at the end of each major topic and are designed to reinforce that topic by permitting the student to apply it immediately. Answers to the in-chapter exercises are provided at the end of the chapter so that the student receives immediate feedback: mastery (or lack of it) is demonstrated while the topic is still fresh in the student's mind. It is strongly recommended that the student do each exercise as it is reached in progressing through the text; in this way, a firm foundation will be laid, block by block, to support the information that follows. The end-of-chapter problems give the student further opportunity to practice the skills gained from reading the chapter and doing the exercises. It should be noted that a special effort has been made to select examples, exercises, and problems that are both realistic and easily visualized by the student. The book is written for the use of the complete beginner with no assumption of business or mathematical experience; yet the examples, wherever possible, have been drawn from or based on real-life programming problems.

The Cobol language can be taught in many different ways, and there are probably as many opinions on the ideal sequence of topics as there are instructors of Cobol. The authors of this text have attempted, insofar as was possible, to avoid imposing their own particular prejudices on the reader. Thus the essentials of the language are presented in Chapters 1 through 4. Chapters 5 and 6 expand upon this groundwork, covering a number of subjects and presenting a number of explanations which would have been unnecessarily confusing had they been presented in the first four chapters. Chapters 7 through 13 treat the "advanced basic" subjects: programmed switches, minor total logic, advanced arithmetic and reporting options, and disk and tape operations. Every attempt has been made to keep these chapters independent so that the user can cover them in any desired sequence. It has been suggested by some reviewers that "Advanced Arithmetic Operations" (Chapter 8) precede rather than follow "More On Conditional Branching" (Chapter 7). There are good arguments for and against such a switch. In order to accommodate the instructor who will be so inclined, care was taken to make the chapters relatively independent of one another so that their order of use can be reversed. In fact, Chapter 7 is divided into two parts to allow the following sequencing if desired: Chapter 7, Part 1; Chapter 8; then Chapter 7, Part 2. Chapter 14 (Modern Programming Techniques) incorporates an example from Chapter 7; the instructor who wishes to teach structured programming may wish to assign this chapter immediately after Chapter 7, or even concurrently with it.

The author of a Cobol text is faced with a problem in selecting a version of Cobol to write about. While Cobol was conceived as a completely machine-independent language, the realities of the marketplace resulted in several distinctive (and non-interchangeable) versions of Cobol. This situation was rectified somewhat in 1968 when a set of standards was published under the auspices of the American National Standards Institute. The 1968

ANS Cobol standard was revised six years later with the publication of the 1974 ANS Cobol standard. In between, various implementors provided their own extensions to the standard; programs using these extensions would not be standard but they would probably be more efficient on a given computer than a standard program. The result, then, is that there are a number of implementations of ANS Cobol, each of which is based on either the 1968 or the 1974 standard, and each of which contains certain unique (or at least non-standard) extensions. In an attempt to make this book useful to as many persons as possible the authors have included the most-used portions of both standards as well as several commonly-used extensions. In every case where a particular feature is not common to both standards, it has been carefully identified. The reader might note that double quotes (ANS standard) rather than single quotes (IBM standard) have been used throughout. This would appear to be inconsistent where IBM Cobol is used as the illustration (for example, when discussing the Environment Division). If the reader uses IBM Cobol, then single quotes should be used throughout or double quotes should be specified in the CBL statement.

This textbook, as are virtually all others, is obviously not the product of solely the authors' minds. The following remark attributed to John Atanasoff, the inventor of the electronic computer, is quite appropriate:

Everything has been around for a long time, you know. Nobody really invents anything. We all lean on the work of others.

And so we have leaned on the work and ideas of our students, colleagues, and fellow authors. In particular, the following have contributed their expertise:

Irwin Kruger, University of Miami
Ben Lewis, San Bernadino Valley College
Gerry Manning, San Francisco State University
Jesse Peckenham, Merritt College

To these individuals and all others who have lent a helping hand, directly or indirectly, we express our sincerest appreciation.

ELEMENTS OF COBOL PROGRAMMING

