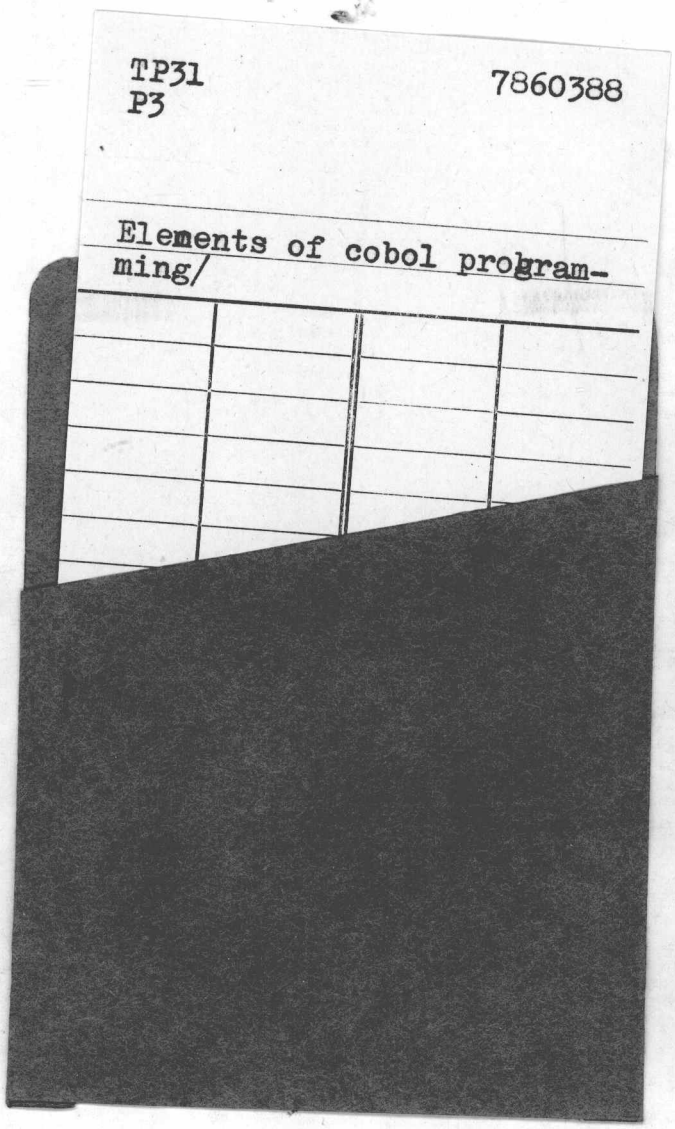
ELEMENTS OF COBOL PROGRAMMING

WILSON T. PRICE
JACK L. OLSON

ELEMENTS OF COBOL PROGRAMMING


TP31
P3
7860388
Elements of cobol program-
ming/

WILSON T. PRICE
JACK L. OLSON

ELEMENTS OF COBOL PROGRAMMING

HRW
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.

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

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 *CO*mmon *B*usiness-*O*riented *L*anguage, better known as *COBOL*.

Cobol, as specified by the CODASYL committee, was far from ideal: it had several drawbacks and a certain cumbersomeness, 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 auspicies 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