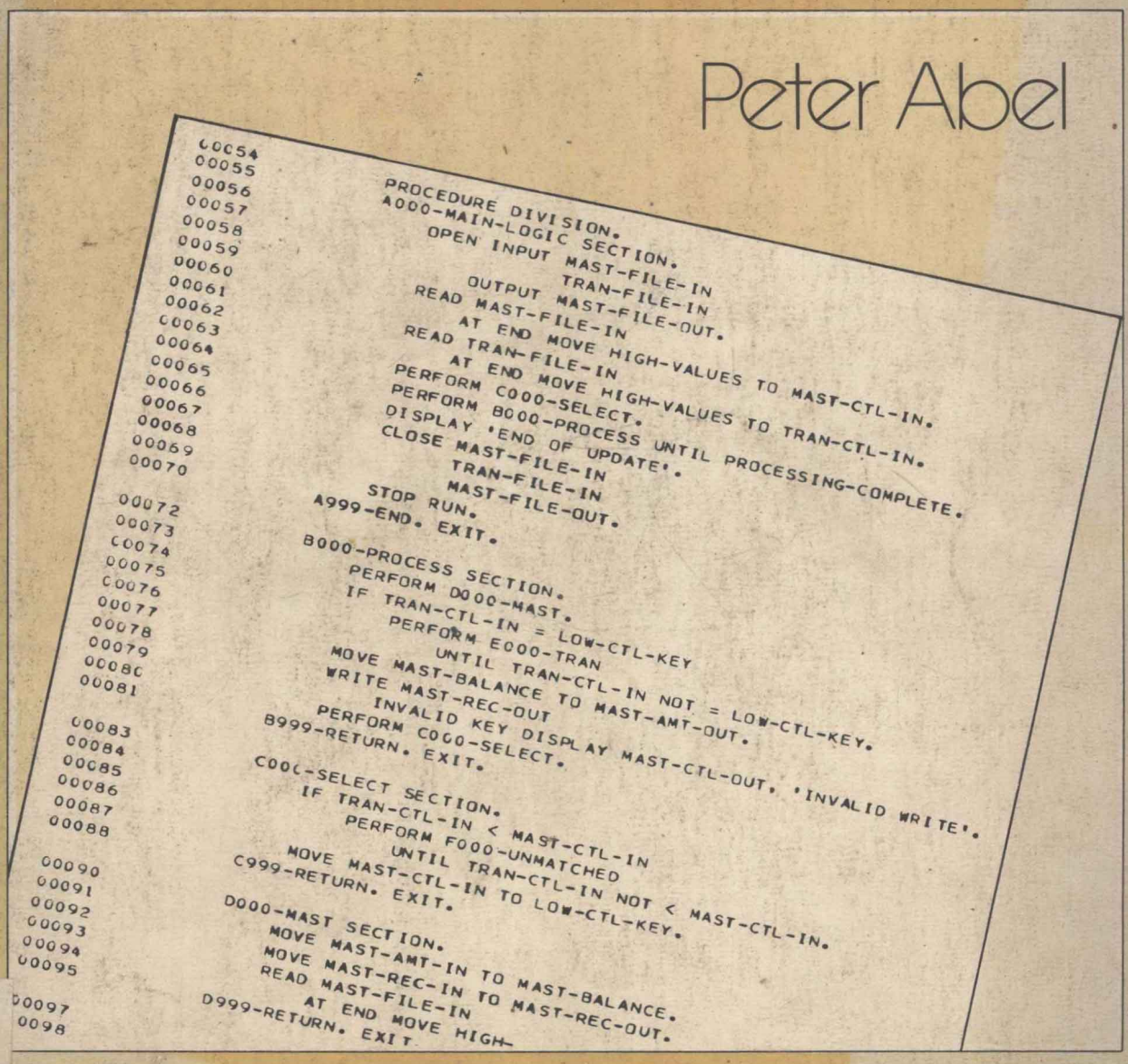
COBOL PROGRAMMING
a structured approach

Peter Abel



```
00054
00055
00056
00057
00058
00059
00060
00061
00062
00063
00064
00065
00066
00067
00068
00069
00070

00072
00073
00074
00075
00076
00077
00078
00079
00080
00081

00083
00084
00085
00086
00087
00088

00090
00091
00092
00093
00094
00095

00097
0098
```

```
PROCEDURE DIVISION.
A000-MAIN-LOGIC SECTION.
    OPEN INPUT MAST-FILE-IN
               TRAN-FILE-IN
         OUTPUT MAST-FILE-OUT.
    READ MAST-FILE-IN
        AT END MOVE HIGH-VALUES TO MAST-CTL-IN.
    READ TRAN-FILE-IN
        AT END MOVE HIGH-VALUES TO TRAN-CTL-IN.
    PERFORM C000-SELECT.
    PERFORM B000-PROCESS UNTIL PROCESSING-COMPLETE.
    DISPLAY 'END OF UPDATE'.
    CLOSE MAST-FILE-IN
          TRAN-FILE-IN
          MAST-FILE-OUT.
    STOP RUN.
A999-END. EXIT.

B000-PROCESS SECTION.
    PERFORM D000-MAST.
    IF TRAN-CTL-IN = LOW-CTL-KEY
        PERFORM E000-TRAN
            UNTIL TRAN-CTL-IN NOT = LOW-CTL-KEY.
    MOVE MAST-BALANCE TO MAST-AMT-OUT.
    WRITE MAST-REC-OUT
        INVALID KEY DISPLAY MAST-CTL-OUT, 'INVALID WRITE'.
    PERFORM C000-SELECT.
B999-RETURN. EXIT.

C000-SELECT SECTION.
    IF TRAN-CTL-IN < MAST-CTL-IN
        PERFORM F000-UNMATCHED
            UNTIL TRAN-CTL-IN NOT < MAST-CTL-IN.
    MOVE MAST-CTL-IN TO LOW-CTL-KEY.
C999-RETURN. EXIT.

D000-MAST SECTION.
    MOVE MAST-AMT-IN TO MAST-BALANCE.
    MOVE MAST-REC-IN TO MAST-REC-OUT.
    READ MAST-FILE-IN
        AT END MOVE HIGH-
D999-RETURN. EXIT.
```

A complete guide to COBOL. Explores DOS and OS usages, gives advanced debugging techniques, and more!

COBOL PROGRAMMING
A STRUCTURED APPROACH

PETER ABEL
British Columbia Institute of Technology

Reston Publishing Company, Inc.
Reston, Virginia
A Prentice-Hall Company

Library of Congress Cataloging in Publication Data

Abel, Peter
Cobol programming, a structured approach.

Includes index.
1. COBOL (Computer program language) I. Title.
QA76.73.C25A23 001.64'24 80-10488
ISBN 0-8359-0833-X

A Prentice-Hall Company
Reston, Virginia 22090

10 9 8 7 6 5 4 3 2

Printed in the United States of America

PREFACE

Programming in COBOL: A Structured Approach offers a new and better approach to COBOL, the most commonly used business programming language. The most elementary features of COBOL are introduced first with a simple example program. After only one chapter of COBOL (Chapter 2), the student is able to write a simple program by concentrating on only the statements necessary to write such a program and by temporarily using the simple ACCEPT and DISPLAY statements for input/output. The text adds new concepts gradually with program examples, and consequently the student is never overwhelmed by the topic.

Structured programming is introduced early. Even before formally discussing the topic, the text adopts and illustrates structured programming features: indentation of code, meaningful names, and organization into main logic and subsidiary sections. By the time Chapter 7 describes structured programming, the student is already accustomed to its features.

A problem in current COBOL texts is omission of specific topics. Thus, one book omits programming style and strategy, another omits control break logic, and another omits advanced handling of tables or disk files. Ideally, one text should cover all of these essential topics, and *COBOL Programming: A Structured Approach* does just this.

Although COBOL is intended to be a universal language, there are two main reasons for the variety of COBOL versions:

1. Differences in computer design (hardware).
2. Differences in operating systems (software).

How does a textbook handle this situation? Because the IBM system is the most common computer system used in the data processing industry, the programs in this text have been designed with IBM features. However, the text points out features that are unique to IBM and that deviate from ANS standards. Your compiler may have some differences from the examples in this text in the following areas:

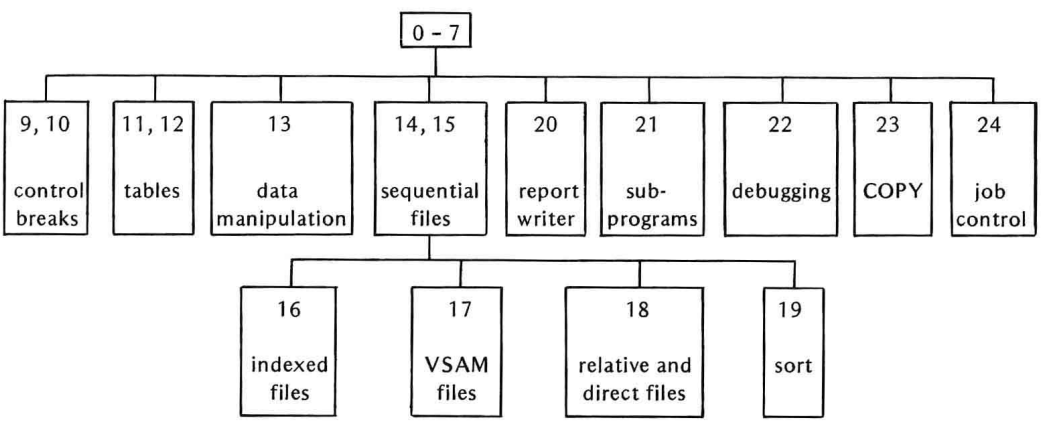
1. Debugging aids.
2. SELECT statements that define input and output files.
3. Definition of arithmetic data in Working-Storage.
4. Specifications for page overflow.
5. Disk devices and disk file organization methods.

These areas represent a small portion of a COBOL program; consequently, students should be able to easily adapt the material in this text to their compiler version. Students should also use the manufacturer's COBOL reference manual along with this text.

In the text, a reference to the IBM 370 or 360/370 implies the general IBM computer series based on the 360. This includes the 3000 series released in 1977 and the 4300 series released in 1979.

ORGANIZATION

Chapters 0 and 1 contain elementary material on computers and programming. Those familiar with another language can skip these chapters or use them as a review. Chapters 2 through 5 cover the basics of COBOL, representing 80 percent or more of standard program coding. Chapters 6 and 7 are concerned with programming style and the use of PERFORM in structured programming. From this point on, the chapters can be studied in various sequences. (Chapter 8, Report Design, is independent of COBOL programming and may be referenced at any time.) The following chart indicates the relationships of the chapters. For example, Chapters 9 and 10 in a box means that Chapter 9 should be read before Chapter 10.

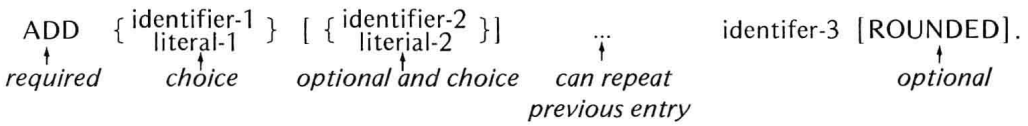


NOTATIONS USED IN THIS TEXT

Some COBOL statements allow a number of options; the text shows their general format according to the following pattern:

Uppercase words	Words in uppercase, such as SPACES and READ, are COBOL reserved words. They must be spelled correctly and used in a program only according to COBOL specifications.
Lowercase words	Words in lowercase are generic terms that represent words or symbols that the programmer supplies.
Brackets [ ]	Brackets indicate that the enclosed entries are optional and may be included or omitted.
Braces { }	Braces enclose two or more vertically stacked items, one of which the programmer is to select.
Ellipses ...	Ellipses immediately following a pair of brackets or braces indicate that the programmer can optionally repeat the enclosed material.

For example:



ACKNOWLEDGEMENT

The author is grateful for the assistance from all those who contributed typing, reviews, and suggestions and to IBM for permission to reproduce some of their copyrighted materials. The following materials are printed with permission, and with modifications, from publications copyrighted in 1972, 1973, 1975, and 1978 by International Business Machines Corporation as IBM form numbers GC20-1649, GC20-1684, GC28-6396, and GA33-1515: Figures 0-3, 8-1, 14-1, 14-7, Appendix B, and general formats of COBOL statements.

PETER ABEL

COBOL PROGRAMMING

CONTENTS

PART II BASIC COBOL

Appendices

PART I

COMPUTER FUNDAMENTALS

CHAPTER 0

INTRODUCTION TO COMPUTERS

OBJECTIVE: To examine the internal characteristics of the digital computer, its input/output devices, and the function of the stored program.

INTRODUCTION

Computers, like other automated devices, were developed to replace human labor. There are two classes of computers, *analog* and *digital*. The *analog* computer measures physical variables such as rotation speed, water pressure, and electric current. Speedometers, steam pressure gauges, and barometers are good examples of analog computers. The *digital* computer works with digits to perform calculations. Unlike the analog computer that can perform only one function, the digital computer can perform many functions through the use of a *stored program,* which is a set of instructions that the computer executes. Another program can replace the stored program at any time to perform a different function. This text uses the term *computer* to mean the common digital computer.

The modern digital computer using the stored program was developed in the 1940s to provide fast, accurate calculations to solve complex problems. Since that time the computer has made its impact in two main areas: scientific and business. In the scientific area, the computer performs calculations for applications such as bridge and building designs, simulation models of the national economy, and statistical studies of the population. In the business area, the computer processes and controls large volumes of data (data processing). Common business applications include accounting, billing, sales analysis, inventory and production control, and airline reservations. In addition, government agen-