COBOL For Beginners

Thomas Worth

THOMAS WORTH

COBOL For Beginners

PRENTICE-HALL, INC., Englewood Cliffs, New Jersey

Library of Congress Cataloging in Publication Data

WORTH, THOMAS (date)
Cobol for beginners.

Includes index.
1. COBOL (Computer program language) I. Title.
QA76.73.C25W67 001.6'424 76-25488
ISBN 0-13-139378-2

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA, PTY. LTD., *Sydney*
PRENTICE-HALL OF CANADA, LTD., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE. LTD., *Singapore*

Preface

This book presents a basic, easy-to-understand introduction to the COBOL programming language. It is intended for classroom use as the central text in a one or two-semester course. It can also be used by persons who must teach themselves COBOL.

The book presents the most useful COBOL features. The details of the language which are less useful or require more student maturity, are deferred in favor of an advanced course or are left for the student to pick up by himself in actual on-the-job training.

With the understanding that a study of COBOL should be undertaken by persons who have a basic knowledge of business practices, the text introduces students to the background knowledge that they must have before they can understand what COBOL can do. In several early chapters, such topics as file organization, the basic data processing cycle, the nature of reports, and others, are briefly explored. Armed with this knowledge, students can make rapid progress in their study of the COBOL language.

The book introduces COBOL from the ground up. No previous experience is assumed. The text presents some information, then reenforces the student's understanding of that information by presenting worked-out programs that use the features discussed. The program's flowcharts are also given.

The problems become more mature as students progress through the text. While not exactly taken from the real world,

the problems discussed are simplified versions of those found there. These problems deal with writing reports, merging files, updating files, and others. Questions and exercises at the end of each chapter, further help strengthen the student's understanding of the various topics and concepts.

This is the kind of book that the author wishes had been available to him when he first learned COBOL. Now, after having taught the subject for several years, he has presented the language in a way that he sincerely hopes will be enthusiastically accepted by those who wish to gain a mastery of the subject.

The author wishes to express thanks to the many persons who have helped develop the text. Among those who gave a great deal of assistance are Karl Karlstrom, Robert Duchacek, JoAnne LaBarge, Freda Allen, and Katie Kid.

Thomas Worth

Contents

CHAPTER	TOPIC	PAGE
1	Introduction	1
2	The Nature Of Files	4
3	Card Files	10
4	Disk Files	17
5	Serial Access File Updating	20
6	Direct Access File Updating	28
7	Reports	34
8	The Basic Data Processing Cycle	39
9	The Steps To A COBOL Program	44
10	A Computer's Memory	59
11	What COBOL Can Do	66
12	The Organization Of A COBOL Program	71
13	The COBOL Coding Form	81
14	Your First COBOL Program	89
15	Job Control Language (JCL)	103
16	Tape File Labels	109
17	The FD Entry	115
18	Levels	122
19	Buffers	127
20	A Report Program	135
21	The MOVE Command	145
22	Editing	155
23	Calculations	170
24	Usage Computational	187
25	Using The WORKING-STORAGE Section	191

CHAPTER	TOPIC	PAGE
26	Making Decisions	203
27	A File Update Program	214
28	The PROCEDURE Division	222
29	Flowcharting	230
30	Multiple Record Files	237
31	Updating Files	256
32	First Time Switch	268
33	Creating A New Master	274
34	Multiple-Page Reports	283
35	Control Breaks	290
36	Tables	307
37	Sorting	317
38	Structured Programming Concepts	335
Appendix	COBOL Reserved Words	352
	Index	355

CHAPTER 1

Introduction

COBOL is a computer programming language. It is used by persons who give instructions to a computer when solving business problems.

The letters in the word COBOL are taken from the words "Common Business Oriented Language". As the name of the language indicates, COBOL is ideally suited for the solution of business problems. For example, if a company wants to set up a payroll system, the instructions to the computer concerning which people are to be paid, when, how much, etc., could well be written in COBOL.

COBOL is often called an "English language" language. The reason is that much of what you write in COBOL, is done using English words. For example, you may use words like DATA, FILE, RECORD, ADD, MOVE, SELECT, READ, WRITE, and many others. These words are part of the COBOL "reserved word" list which we will present shortly.

Reserved words must be used in exactly the way that COBOL demands that they be used. With only a few exceptions, each word has only one meaning. In this text we will discuss many of the reserved words and tell what their meanings are. We will also give examples of how those words may be used. Take a moment now to glance at the appendix that contains the reserved words used in COBOL.

Persons who use COBOL may invent words of their own. For example, a programmer may invent words like PAY-RATE, YEAR-TO-DATE-EARNINGS, PERSONNEL-MASTER-FILE, and many others. There

is an important restriction. A user must make sure that invented words are not on the reserved word list. We'll show you how to invent your own English words later in this text.

When you use COBOL, you are performing a function called "programming". Programming refers to the writing of instructions for a computer to follow in solving a problem. The set of instructions is called a "program". Reviewing, a program is a set of instructions to a computer telling it how to solve a given problem. The process of creating a program, is called "programming". A person who creates programs, is called a "programmer".

There are other programming languages available besides COBOL. Some of those languages are FORTRAN, PL/I, RPG, ALGOL, and BASIC. Each of the languages has its particular strength; for example, FORTRAN is especially useful for the solution of scientific problems. This text discusses COBOL, an excellent business programming language.

This book is for beginners. The author assumes the reader has had no previous experience in data processing. We begin at the beginning and proceed in easy steps toward a mastery of the major features of COBOL. Not all features of COBOL are discussed in this book. To do so within the step-by-step framework of this text would require a much larger volume. However, the text covers the most useful features. In addition to the material covered in this text, you will also be given instructions on how to understand COBOL reference manuals. You can use the reference manuals to augment your knowledge of COBOL. Once you know how to code simple COBOL programs, it is but a single step toward the expertise you need to write more complex programs.

We trust you will have an enjoyable and rewarding experience as you move forward to the remaining chapters in this book. Good luck!

EXERCISES AND PROBLEMS

1. What do the letters in the word COBOL stand for?
2. Why is COBOL called an English-language language?
3. What is the COBOL "reserved word" list? Give six words that are in the list.
4. What is a program?
5. May programmers use invented words, as well as COBOL reserved words, in COBOL programs?
6. Besides COBOL, what are the names of some other programming languages?

NOTES FROM THE PROGRAMMER'S DESK

A computer is very fast. It can perform in a single second the amount of work that a human being would require one full year to duplicate.

But a computer is not a magical device. It has to be told *what* to do and in *what sequence*. If a person gives incorrect instructions to a computer, then the computer *must* give incorrect results. Those incorrect results will be given at amazing speed.

As you begin your study of computer programming using the COBOL language, resolve to be thorough and accurate. It has been said, and accurately so, that garbage in equals garbage out (GIGO). Avoid GIGO because GIGO is expensive.

CHAPTER 2

The Nature of Files

COBOL is "file-oriented". This means that a COBOL programmer, using the language, can describe files, read files, create files, and process the data found upon files. (We'll discuss the meanings of "data" and "data processing" presently.)

Since COBOL is so oriented toward files, it follows that before proceeding with anything else, we should discuss files, determine what they are, and study how they can be used. There are all sorts of files. Let's consider business files.

A business file is a collection of related "data records". Records are made up of "fields" and fields are made up of "character positions". To understand the meanings of these terms, let's consider a concrete example.

In the Sales Department of the Wilcox Manufacturing Corporation, there are 230 employees. The business facts that the company needs for each employee are recorded on a reel of magnetic tape. Figure 2-1 on the following page shows that tape.

Another word that means the same as "facts" is "data". From this point we will use the term data to represent business facts. (Observe that the word "data" is plural.)

The reel of tape shown in Figure 2-1, and the data recorded upon it, constitute an entity called a "master file". A "master file" contains *all* of the data for *all* of the related persons or things that the file covers. As an example, if a file is an inventory master file, the file contains *all*

of the data for *all* of the parts in inventory. If the file is a personnel master file, the file contains *all* of the data for *all* of the employees of the business unit.

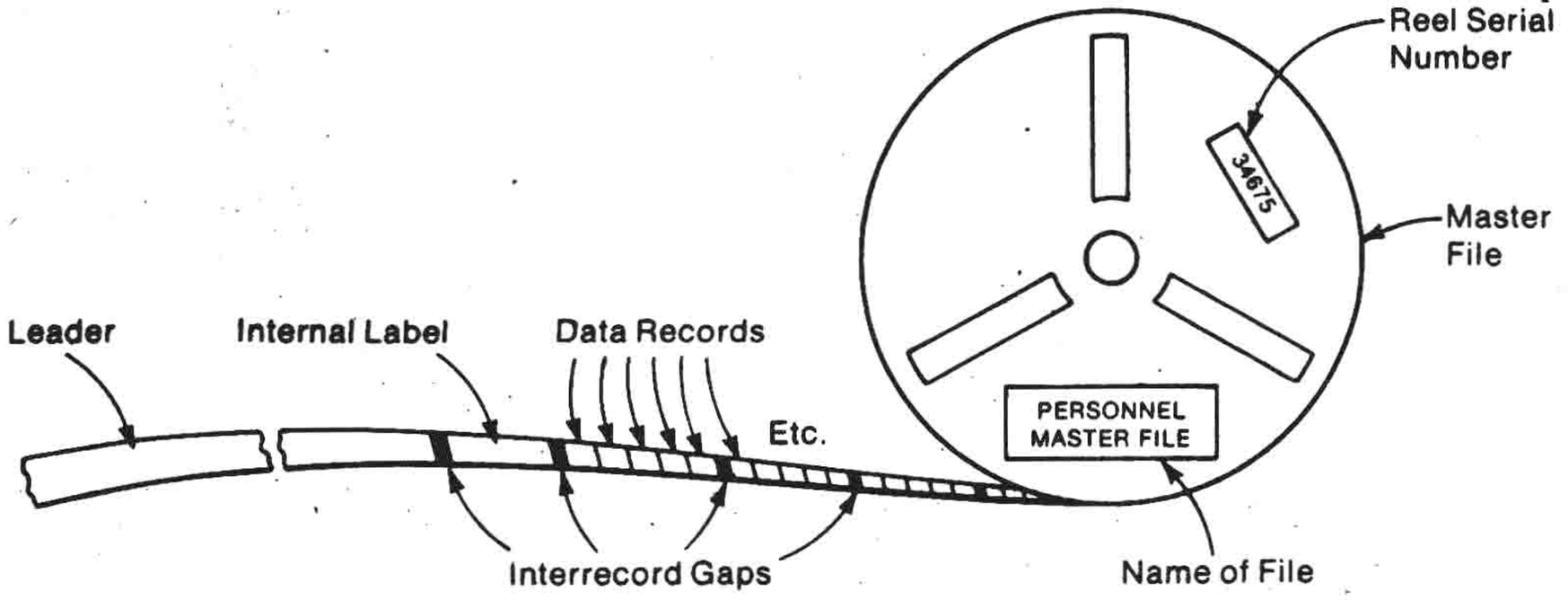

Figure 2-1

Typically, a reel of tape is 2400 feet long and 1/2 inch wide. By means of magnetized spots on the surface of the tape, as many as 6250 characters of data may be recorded *per inch*. It is common to find tape with a recording density of 200, 556, 800, and 1600, as well as 6250, characters per inch.

In Figure 2-1 you see a "leader". The leader may be up to twenty feet long. Its function is to provide some tape that can be wound on a takeup reel on the device used to process tape (a tape handler). Figure 2-2 on the following page shows a tape handler.

On the tape shown in Figure 2-1, you can see unused portions. Each unused portion is a 3/4" blank area. The blank area is called an "interrecord gap". Soon we'll discuss why interrecord gaps exist.

You can also see an "internal label". An internal label is the name given to 84 characters of data. The characters tell what is the name of the file (for example, Employee Master), the reel serial number (for example, 34675), the date created (for example, March 28, 1977), and other items. As we

shall see when we discuss the "opening" of input files, internal labels are used by computers to check files before they are used.

From De Rossi, *Exploring the World of Data Processing* c 1975, Reproduced by permission of Reston Publishing Company, Inc, Reston, Virginia

Figure 2-2

Between interrecord gaps we see groups of five records. A record is a collection of all the data items concerning *one* individual. For example, the first record of the illustrated file may be George T. Milliken's record. In this record, George's social security number is recorded; also his name, job code, pay rate, and year-to-date earnings. Several other data items concerning George Milliken are also given.

The next record may be Helen G. Bradley's record. Here, Helen's social security number is given; also her name, job code, pay rate, year-to-date earnings, etc.

The data items mentioned above: social security number, pay rate, job code, etc., are recorded on the tape in places called "fields". A field is an area made up of character positions. For example, an area on the tape may have been reserved to hold a social security number, another area to hold an employee name, another to hold a job code, etc. The areas are of variable length because some data items require more character positions than others.

The smallest unit on a tape is the character position. A character position is capable of holding a single data processing character such as A, 3, $, etc. These characters are stored in the form of magnetized spots. One or more character positions form a field; one or more fields form a record; and one or more records form a file.

The records on a personnel master file are organized around some key data item. Following common business practice, the key item is probably each individual's social security number. The records are probably recorded in increasing sequence by social security number. The records on an inventory master file are probably recorded in increasing sequence by part or catalog number.

In Figure 2-1, observe that there is an interrecord gap at the end of the fifth record. The records in the group of records constitute a unit called a "block". Observe that there is a group of five records per block in the illustrated file.

A block of records may contain one, two, three, or as many records as was deemed desirable by the person who designed the format of the file.

For the present, let us assume that each record in the file is exactly the same size as every other record. This organization necessarily means that some space in each record is wasted because not all data items within the records require the same amount of space (in character positions). The name "Milliken", for example, is somewhat longer than the name "Bradley". If each record has reserved 15 character positions for a last name, Milliken's record uses eight of the 15 positions; Bradley's record uses only seven. The unused character positions receive blanks. Similarly, if seven character positions have been reserved for year-to-date earnings, one person's record may use only ~~six~~ of them, another record, five. Where numeric values are concerned, unused positions in a record are filled with zeroes.

Let's review:

A file is a collection of data records. Often, but not always, all the records have been designed to be of the same size.

The person who designed the file decided how many records to place in a "block". Interrecord gaps separate blocks of records.

Records consist of fields. A field contains a data item like social security number, employee name, job code, pay rate, year-to-date earnings, year-to-date federal taxes paid, number of exemptions, etc.

Fields consist of character positions. For example, social security numbers require nine positions (dashes are not included in storage), last names require 15 (more or less) character positions, year-to-date earnings require seven character positions, etc.

At the head of the file, an "internal label" gives information about the file so that the file can be checked before it is used.

On magnetic tape, records are stored in blocks. A block may contain from one to many records. When the computer is directed to obtain a record from the file, it obtains, not one, but an entire block of records. The computer reads (obtains) records from one interrecord gap to another. We'll have more on this subject later.

Magnetic tape can be read at rates of 120,000 characters per second and more. This means that all the data on a reel of magnetic tape can be read in three minutes or less.

EXERCISES AND PROBLEMS

1. What is a file?
2. What is a record?
3. What is a field?
4. What is a character position?

5. What is a word that means the same as "data"?

6. How long, in feet, is a typical reel of tape? What is its width?

7. How many characters per inch may be recorded on magnetic tape? What are some common recording densities?

8. What is the function of a "leader" on magnetic tape?

9. What is an "interrecord gap"?

10. What is an internal label? What is the function of the label on magnetic tape?

11. How many records would you expect to find in a block on magnetic tape?

12. What is a "key" data item?

13. Do records in a file always have uniform sizes? Explain.

14. What does the term "read" mean? How many records does a computer read when requested to obtain one?

15. Fill in the blanks: A file is a collection of _____. A record is made up of _____. Fields consist of _____ _____.

16. If all 2400 feet on a reel of magnetic tape were usable for data, how many characters of data could the tape hold? Assume a recording density of 800 characters per inch.

17. A file on magnetic tape contains 5000 records. Each record contains 3000 character positions. There are ten records per block. How many feet of tape are needed to record the file? Assume that the interrecord gaps are 3/4" long and that the recording density is 800 characters per inch. Ignore the tape leader and the internal label.

18. Solve the same problem given in Question 16. There is only one change in the facts given. Assume that the recording density is 6250 characters per inch.

CHAPTER 3

Card Files

Master files are normally stored on magnetic tape, magnetic disk, or data processing cards like the one shown in Figure 3-1.

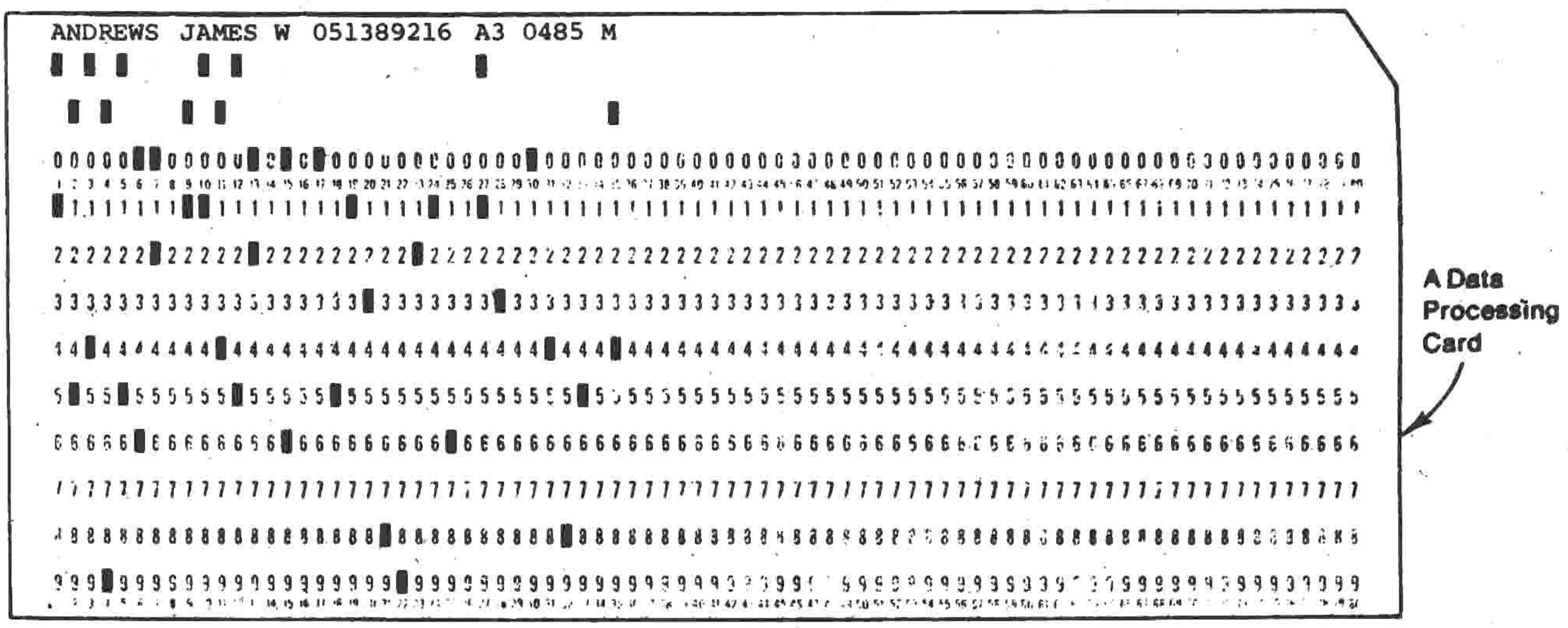

Figure 3-1

The illustration shows one punched card. Consider this card to be James Andrews' record. You can see that the card contains Andrews' name (ANDREWS JAMES W), social security number (051389216), job code (A3), pay rate (0485), and sex (M). Many master records contain much more data than that shown in the illustration.

Now consider a deck of master record cards. See Figure 3-2 on the following page.