Eastern Economy Edition

V. RAJARAMAN
H.V. SAHASRABUDDHE

COMPUTER PROGRAMMING IN COBOL

PHI

Computer Programming in COBOL

V. RAJARAMAN
Ph.D.
Professor of Electrical Engineering and Computer Sciences
Indian Institute of Technology
Kanpur

H. V. SAHASRABUDDHE
Ph.D.
Professor of Computer Sciences
Indian Institute of Technology
Kanpur

Prentice-Hall of India Private Limited
New Delhi-110001
1981

Rs. 39.00

COMPUTER PROGRAMMING IN COBOL
by V. Rajaraman and H. V. Sahasrabuddhe

PRENTICE-HALL INTERNATIONAL, INC., Englewood Cliffs.
PRENTICE-HALL OF INDIA PRIVATE LIMITED, New Delhi.
PRENTICE-HALL INTERNATIONAL, INC., London.
PRENTICE-HALL OF AUSTRALIA, PTY. LTD., Sydney.
PRENTICE-HALL OF CANADA LTD., Toronto.
PRENTICE-HALL OF JAPAN, INC., Tokyo.
PRENTICE-HALL OF SOUTHEAST ASIA (PTE.) LTD., Singapore.
WHITEHALL BOOKS LIMITED, Wellington, New Zealand.

ISBN-0-87692-030-10

Photocomposed and printed by Mohan Makhijani at Rekha Printers Private Limited New Delhi—110020
and Published by Prentice-Hall of India Private Limited, M—97, Connaught Circus, New Delhi—110001.

"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 CODASYL Programming Language 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.

"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 Automations 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."

Preface

This book introduces the basic concepts of programming a computer for business application. The emphasis is on problem-solving using COBOL (Common Business Oriented Language). The presentation is from first principles for readers with no prior knowledge of computers. Programming is introduced in gradual steps with a large number of worked examples. Numerous programs have been included in the book and all of them have been run on a DEC-1090 computer at the Indian Institute of Technology, Kanpur. The actual programs have been photo-offset to prevent typographical errors.

A distinctive aspect of the book is the emphasis on good style in COBOL programming we have placed throughout all the discussions. We have also illustrated a methodology of step-wise development of programs and good programming style through many programs developed in this book.

The COBOL used in this book is the standard COBOL as proposed by the American National Standards Institute in 1974 (ANSI 74 COBOL). We have not discussed exhaustively all features of COBOL. Such a discussion would have distracted us from our basic aim of presenting COBOL from first principles for beginners. Commonly used advanced features have, however, been included to make the book useful for a practising programmer. Specifically, we have discussed tables, report generator module, SORT/MERGE feature and indexed sequential file processing.

A book of this type naturally made use of a number of ideas from previous books on this subject. We thank all these authors, too numerous to acknowledge individually. We would like to thank all our colleagues in the Computer Centre at the Indian Institute of Technology, Kanpur, for their ready cooperation. We thank Sri Sankar Iyer, B.Tech. student at I.I.T., Kanpur, for reading the earlier portions of the manuscript and assisting in improving the clarity of presentation. Sri Arjun Puri, a data processing consultant of Delhi, gave valuable comments on the earlier chapters and we thank him for his help. The assistance of Sri Pradeep Jain, M.Tech., Computer Science student who critically read the entire manuscript and suggested improvements, is gratefully acknowledged. We thank Sri H. K. Nathani for prompt and accurate typing of the manuscript and Sri. A. K. Bhargava for advice on line drawings.

V. Rajaraman

H. V. Sahasrabuddhe

Kanpur:
August 27, 1981.

Computer Programming in COBOL

Contents

1 What Are Computer Programs?

Writing a program for a computer is like giving detailed orders to a clerk to do a job. The basic assumption we make is that the clerk obediently and literally follows orders. We assume that he has no "commonsense". We will illustrate with an example.

Example

It is required to give a set of instructions to a clerk on how to issue items from a store. When a customer asks for an item from the store the clerk is to check if it is available in the store and give an issue slip.

The first step in making up the instructions (to be followed by the clerk) is to decide how the catalog of items in the store will be organized, in what form the request will be presented and in what form the issue slip will be made.

We will decide to make the catalog in the following manner:

Each item in the store will be given a unique identification number. We will call it the item code. A list will be made with item code, item name, and quantity in stock. For easy reference and retrieval this list may be listed in ascending order by item code. Such a list or catalog is also known as a *master file*. An example of such a catalog is shown in Table 1.1.

TABLE 1.1
PORTION OF A STORE CATALOG

Item code	*Item name*	*Quantity in stock*
0005	RESISTOR 5 K	250
0022	RESISTOR 22 K	152
0348	POTENTIOMETER 1 K	28
0386	POTENTIOMETER 1 M	52
1234	CAPACITOR 1 MF	158
1268	CAPACITOR 10 MF	125
1779	INDUCTOR 1 MH	52

We will make the customer's request form as follows:

For each item needed the customer is to make a request slip. Sample request slips are shown in Table 1.2.

The slip has the item code, item name, quantity required and a blank column in which the quantity issued is to be entered by the clerk. Observe that the customer should know the unique code number assigned to each item in the catalog.

The clerk takes the request slips from the customer and fills up the quantity issued column. Sample issue slips corresponding to the request slips of Table 1.2 are shown in Table 1.3.

Having decided the form in which data are organized and the manner in which results are to be presented, the next step is to evolve a set of detailed specific instructions to the clerk to perform the job, assuming that the clerk does not have any commonsense but will obey faithfully all orders given to him. The instructions are as follows:

Procedure to issue items from store

Step 1: For each request slip presented by the customer do step 2 to step 8. When request slips are exhausted, stop.
Step 2: Read item code from request slip.
Step 3: Compare item code in request slip with item codes in catalog serially starting from the first line in the catalog.
Step 4: If there is a match go to step 6.
Step 5: If there is no match then enter 0 (zero) in quantity issued column go to step 8.
Step 6: Compare quantity requested with the quantity in stock (as given in catalog).

Step 7: If the quantity requested is less than or equal to the quantity in stock *then* copy quantity requested from the request column to the issue column, subtract the quantity issued from the quantity in stock and enter the result as the quantity in stock; *else* enter quantity in stock (as found in the catalog) as quantity issued in the issue slip, enter zero as quantity in stock in the catalog.

Step 8: Return issue slip to customer.

Observe the large number of detailed instructions given above to do this simple job. This much detail is necessary as we assumed that these instructions are to be followed by a clerk without commonsense. A computer is a machine which has no "commonsense" but will faithfully and literally follow instructions. Thus detailed instructions, such as those given above, are required for solving problems by computers.

TABLE 1.2
CUSTOMER'S REQUEST SLIPS

Slip 1

Item code	*Item name*	*Quantity required*	*Quantity issued*
0022	RESISTOR 22 K	10	

Slip 2

Item code	*Item name*	*Quantity required*	*Quantity issued*
1238	CAPACITOR 2 MF	10	

Slip 3

Item code	*Item name*	*Quantity required*	*Quantity issued*
1268	CAPACITOR 10 MF	150	

TABLE 1.3
ISSUE SLIPS RETURNED BY STORES CLERK

Issue Slip 1

Item code	*Item name*	*Quantity required*	*Quantity issued*
0022	RESISTOR 22 K	10	10

Issue Slip 2

Item code	*Item name*	*Quantity required*	*Quantity issued*
1238	CAPACITOR 2 MF	10	0

Issue Slip 3

Item code	*Item name*	*Quantity required*	*Quantity issued*
1268	CAPACITOR 10 MF	150	125

A number of important features of computer programs are brought out by the above example. They are:

1. A program consists of a finite list of detailed instructions to be followed exactly in the given sequence.
2. Each instruction must be precise and lead to a definite action.
3. Before a computer oriented procedure can be evolved it is necessary to decide the form in which the results and input data will be presented. In the example discussed the structure of the catalog, the request slips and the issue slips were decided before writing the procedure.
4. The procedure should take a finite time to execute and should come to a halt.
5. There are two basic types of instructions in a computing procedure. The first type is data oriented. Data movements, comparison and arithmetic instructions such as addition, subtraction, multiplication and

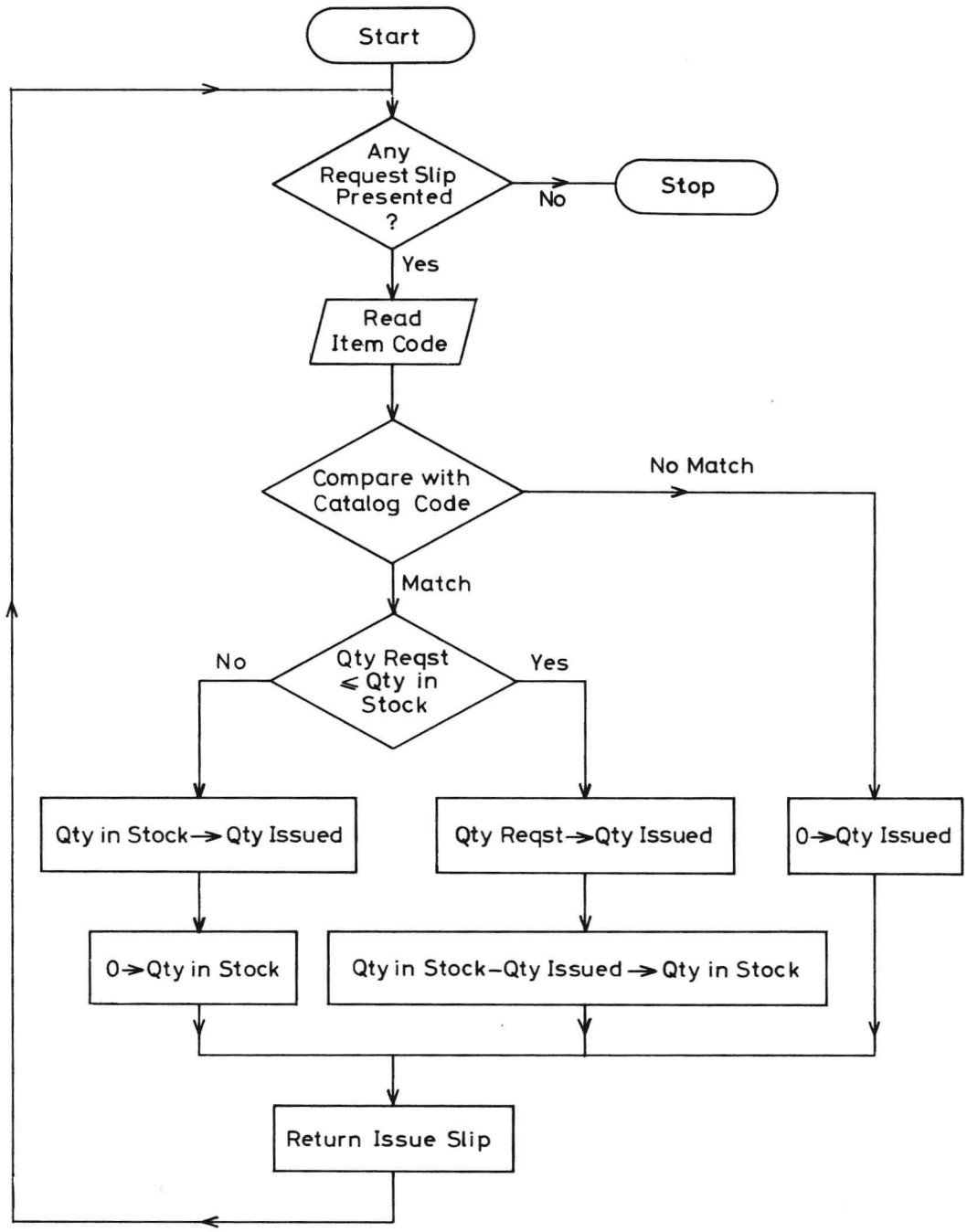

Fig. 1.1. Flow chart illustrating stores issue.

division are in this category. The second type called control instructions determine the sequence of steps to be followed in a procedure. In the example presented above steps 3 and 6 are examples of data operations. Step 4 is a control instruction. Step 7 combines data operation with control (sequencing) operation.

6. The structure of a procedure written for solving a problem on a computer does not change when the input data changes. In the example considered the set of instructions given to the clerk would not change if the catalog changes.

It would make our life easy if raw data for a problem could be fed to a computer and the computer could on its own analyse the problem, arrive at the correct method of solution, follow the method and produce the right answers. ***No computer can do this***. We cannot give a computer *vague* instructions about what we want it to do. We must clearly specify by means of precise instructions what we want the computer to do and in what order they are to be done.

The detailed instructions given in the example are lengthy and difficult to grasp quickly. As an aid to formulating such instructions a pictorial representation called a flow chart is often used. A flow chart for the example is given as Fig. 1.1. This illustrates the value of a picture. In the next chapter we will discuss in detail the development of flow charts.

EXERCISES

1.1 In the procedure given in this chapter for issuing items from a store it is assumed that if the item code in a request slip is not in the catalog a 0 is entered in the issue column. Modify the procedure to create a separate file containing those requested items not found in the catalog.

1.2 In the procedure for stores issue if the quantity requested by a customer is available partially (for example, if a customer requests a quantity of 10 and only 5 is available) then the partial quantity available is issued. Modify the procedure so that if only partial quantity is available the item is not issued.

1.3 In the procedure for stores issue it is assumed that the items in the catalog are arranged in ascending order of item codes. Why is it necessary to assume this? What would be the problem if such an arrangement is not used?

1.4 Evolve a procedure to instruct a clerk how to make a bill of sale when items are sold. Clearly define the format of the data required for implementing this procedure.

1.5 Evolve a procedure which a bank clerk should follow to cash a self cheque presented to him by a customer.

1.6 Take a woman's magazine such as *Femina*. Look up a recipe for cooking. Express the instructions as a step-by-step procedure.

1.7 Evolve a procedure to be followed by a customer in a bank to keep his pass book up-to-date.

1.8 A teacher holds a test and marks papers out of 100 marks. Evolve a step-by-step procedure he should follow to create a list of students who score less than 40 marks in the test.

2 Flow Charts

Before a computer can be used to solve a problem it is necessary to plan the strategy to solve it. This planning is aided by using *flow charts*. A flow chart is a picture which shows the sequence in which data are read, computing performed, decisions made and results presented. Flow charts are useful both for planning the strategy of the solution to a problem and for documenting the method used. In this chapter we will show with a number of examples how one develops a flow chart to solve problems.

2.1 A flow chart

Example 2.1

A company has salesmen selling two varieties of cloth, namely, cotton and terecot. The company gives 6 per cent commission on sales of cotton cloth and 5 per cent on terecot. It is required to compute the commission earned by a salesman.

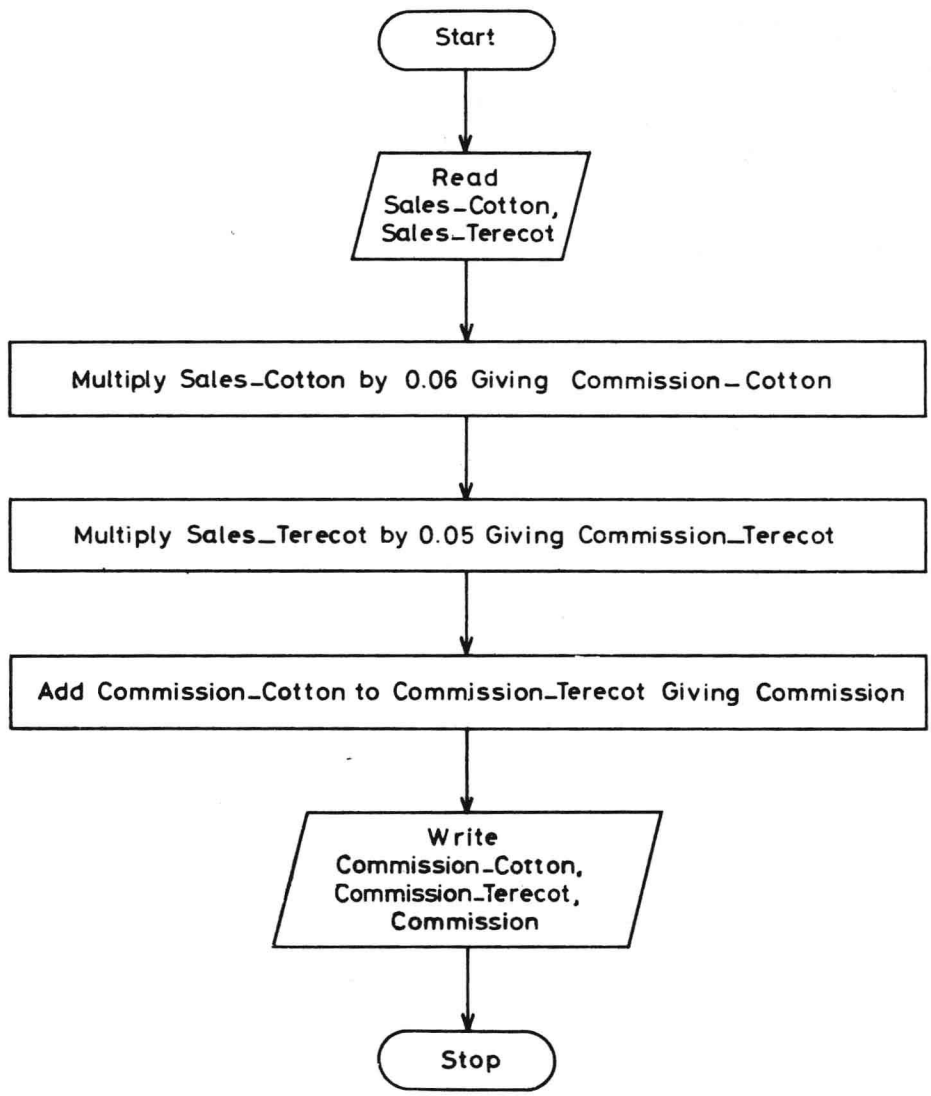

Fig. 2.1. Flow chart for computing a salesman's commission.

Let SALES-COTTON be the cotton cloth sales and let SALES-TERECOT be the terecot sales. Then the commission on cotton sales is obtained by multiplying SALES-COTTON by 0.06 and the commission on terecot sales is obtained by multiplying SALES-TERECOT by 0.05.

The steps in the solution are:

Step 1: Read the values of SALES-COTTON and SALES-TERECOT.
Step 2: Multiply SALES-COTTON by 0.06 giving COMMISSION-COTTON.
Step 3: Multiply SALES-TERECOT by 0.05 giving COMMISSION-TERECOT.
Step 4: Add COMMISSION-COTTON to COMMISSION-TERECOT giving COMMISSION.
Step 5: Write the values of COMMISSION-COTTON, COMMISSION-TERECOT, COMMISSION.
Step 6: Stop.

These steps are pictorially shown as a flow chart in Fig. 2.1 The shapes of various blocks used in the flow chart are the same as those recommended by the International Standards Organization.

The first block in the flow chart, labelled START, signals the beginning of the procedure. A rectangle with rounded edges is used to represent START, and STOP operations. The next operation is to read the values of SALES-COTTON and SALES-TERECOT. In the flow chart a parallelogram is used to represent reading of data. A line with an arrow connects any two blocks in the flow chart. The arrow indicates the order in which operations are to be carried out. Beginning at START, we follow the arrow and carry out subsequent operations. In the next block in the flow chart we compute COMMISSION-COTTON. Computational operations are enclosed in rectangles in a flow chart. The block following the three rectangles is a parallelogram in which is written WRITE COMMISSION-COTTON, COMMISSION-TERECOT, COMMISSION which states that their values are to be printed. The last block in the flow chart is a command to stop.

The other symbols used in a flow chart are shown in Fig. 2.2. These are:

A *decision symbol* (a diamond-shaped symbol) which is used to indicate a decision being taken and the paths to be followed based on the answer to a question or the result of a comparison.

A *connector symbol* (a circle) which is used to connect together portions of a flow chart split between pages. A

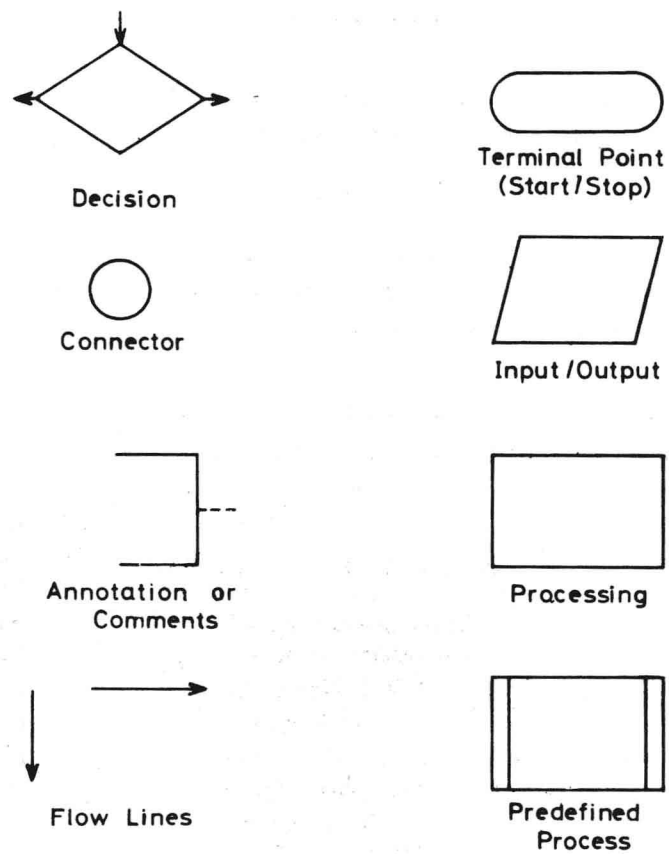

Fig. 2.2. Flow chart symbols.