COBOL PROGRAMMING

M. K. ROY • D. GHOSH DASTIDAR

COBOL Programming

M K ROY
D GHOSH DASTIDAR
Jadavpur University
Calcutta

Tata McGraw-Hill Publishing Company Limited
NEW DELHI

Sponsoring Editor : Rajiv Beri
Editorial Assistant : Ranjan Kaul
Production Assistant : Anson Babu

Published by Tata McGraw-Hill Publishing Company Limited, 12/4 Asaf Ali Road, New Delhi 110002 and printed at Rekha Printers Private Limited, A 102/1, Okhla Industrial Estate, Phase II, New Delhi 110020.

McGraw-Hill Offices

NEW DELHI

New York
St Louis
San Francisco
Auckland
Bogotá
Guatemala
Hamburg
Lisbon
London
Madrid
Mexico
Montreal
Panama
Paris
San Juan
São Paulo
Singapore
Sydney
Tokyo
Toronto

To our Parents

PREFACE

This book is intended for those who are interested in COBOL. It can well serve as a textbook at the undergraduate level where COBOL is included as a part of the curriculum. It can also be used by the participants of the COBOL courses offered under the Continuing Education Programme of the various Indian universities and institutions. Programmers and other computer professionals will also find the book helpful.

A knowledge of computers is not a prerequisite to follow the contents of the book. The various introductory concepts on computer systems and programming are discussed in the first two chapters. The basic aspects of the COBOL language have been introduced in the next five chapters. These have been presented in such a way that having gone through these five chapters, the reader would be able to write, at least, simple COBOL programs. Further details of the basic features as well as other essential features are covered in Chapters 8 to 14. The rest of the book deals with advanced topics.

We faced some difficulty in selecting the material for this book, because there exist different versions of COBOL. The 1974 American National Standard COBOL (called ANSI 74 or ANS 74 COBOL), which is the more recent version of COBOL, includes a number of incompatibilities with the previous version, namely, ANSI 68 COBOL. Most of the compilers that implemented ANSI 68 have been subsequently modified to incorporate some new features but they still retain some old features that are incompatible with ANSI 74 COBOL. Moreover, the ANSI standards leave many features to be defined and implemented by the implementors. As such, in this book, we have taken a middle course. We have assumed a typical implementation of our own and have described the features accordingly. In addition, at the end of each relevant chapter we have discussed the implementation differences of three COBOL compilers, namely, those for B-6700, DEC-10 and ICL-1900. This will help the reader in understanding the nature of differences that exist between one implementation and another. As regards the typical implementation of the text, the description has been based on carefully selected features of ANSI 74 and those of the said three implementations. Moreover, two other implementations, namely, IBM 370 COBOL and NCR VRX COBOL have also been taken into consideration for the said purpose. However, we have not included the implementation differences of these latter named versions, because information regarding IBM 370 COBOL is easily available elsewhere and most of the features as described here do not have much differences with the NCR VRX COBOL. In fact, we find that most of the ANSI 74 features have been faithfully implemented in NCR VRX COBOL.

To avoid any misinterpretation, we would like to spell out clearly the purpose of including implementation differences. The objective is to indicate to the reader where an implementation difference may exist and what its nature may be. The notes given under implementation differences, therefore, point out only some of the important and interesting differences. They certainly do not reveal all differences nor do they show the relative merits and demerits of the said compilers. Moreover, the information given under implementation differences are subject to change as the compilers are normally modified frequently. As such, the readers should not take these notes too seriously. In order to make use of implementation dependent features, one must consult the relevant manuals.

It will be wrong to assume that COBOL is an implementation dependent language. In fact, there are more similarities than differences. The differences arise only when one wants to make specific use of the language elements. A knowledge of implementation differences helps one to avoid writing the implementation dependent code. This makes a program portable so that it can be executed on different machines with minimum change.

COBOL is a powerful language and offers a lot of facilities. The power of COBOL cannot be fully exploited if one learns only a subset of COBOL that enables him to write an application program. This book, therefore, attempts to present the language features with as many details as may be useful. The book also covers a substantial part of ANSI 74. In fact, except for the debugging and communication modules of ANSI 74, all other language features have been included.

We thankfully acknowledge the help that we have obtained from the various published materials, a list of which is given in the bibliography. We are indebted to many of our friends and colleagues with whom we have had discussions regarding the contents and structure of this book. Notable among them are Mr. T.K. Basu of ICL (India) Pvt. Ltd., Mr. A. Banerjee, Mr. A.K. Majhi, Dr. Subrata Roy and Mr. D.P. Sinha. Our special thanks are to Sri Priyatosh Chakraborty for typing the manuscript and to Sri Samaresh Bhattacharya for drawing the figures.

Finally, we would like to include an apology to those readers who belong to the fair sex. In the text we have referred to the programmer or reader as "he". This may be taken to mean "he or she".

ACKNOWLEDGEMENTS

The following acknowledgement is reprinted from the American National Standard Programming Language COBOL, X3.23-1974 published by the American National Standards Institute, Inc.

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. F 28-8013, 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.

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.

M K Roy
D Ghosh Dastidar

CONTENTS

1

INTRODUCTION TO COMPUTER SYSTEMS

1.1 DATA PROCESSING

Modern computers can work at very high speeds and at the same time are very reliable. The work that is normally done by a computer is called data processing. Note that there are two words, data and processing. Data are entities that relate to a certain person, task or event. Processing means performing systematic operations upon the data to make them more useful. Such manipulations that a computer can perform on the data include calculation, comparison, reading and writing. Calculation refers to the execution of arithmetic operations, such as addition, subtraction, multiplication and division. Besides calculation, a computer can compare two data to determine which one is greater or to establish whether they are equal or not. Based on the result of the comparison it can also choose one of the two courses of action. Reading is the process of accepting data from some external medium and writing is the reverse process of reading. The results of calculations can be written by the computer onto some medium such as a sheet of paper.

The operations mentioned above are only elementary operations. By combining such elementary operations it is possible for the computer to perform processing that is complex in nature. However, the primary objective of data processing is to take the help of the computer to process some given data in a desired manner. Thus a computer can be called a data-processing equipment which accepts some data as its input and produces some "processed data" as its output. The processed data is often called information.

It is of importance to note that a computer cannot do anything on its own. It must be instructed to do a desired processing. That is why it is necessary to specify a sequence of operations that a computer must perform to solve a data processing problem. Such a sequence of operations written in a language that can be understood by a computer is called a program. It is the program that controls the activity of processing by the computer and the computer performs precisely what the program wants it to do. To enable a computer to execute a program automatically it is necessary that the program must be stored somewhere within the computer in advance. This prestoring of the program is a fundamental requirement and this is why a computer is often referred to as a stored program computer.