STRUCTURED COBOL

A.S. Philippakis and Leonard J. Kazmier

STRUCTURED COBOL
Second Edition

A. S. Philippakis
Arizona State University

Leonard J. Kazmier
Arizona State University

McGraw-Hill Book Company
New York St. Louis San Francisco Auckland
Bogotá Hamburg Johannesburg London
Madrid Mexico Montreal New Delhi Panama
Paris São Paulo Singapore Sydney Tokyo Toronto

Library of Congress Cataloging in Publication Data

Philippakis, Andreas S
Structured COBOL.

Includes index.
1. COBOL (Computer program language) I. Kazmier, Leonard J., joint author. II. Title.
HF5548.5.C2P54 1981 001.64'24 80-12647
ISBN 0-07-049801-6

STRUCTURED COBOL

567890DODO898765

This book was set in Optima by Cobb/Dunlop Publisher Services Incorporated.
The editor was Charles E. Stewart; the production supervisor was Donna Piligra.
R. R. Donnelley & Sons Company was printer and binder.

See Acknowledgment on page xv. Copyrights included on this page by reference.

PREFACE

COBOL (COmmon Business Oriented Language) has been adopted as the standard programming medium for administrative applications of the computer in industry and government. The language has been continuously reviewed and modified under the sponsorship of the Conference on Data Systems Language (CODASYL), a working committee composed of representatives from major computer language users and computer manufacturers. The current version of COBOL was adopted as the standard by the American National Standards Institute (ANSI) in 1974 and is a revision and refinement based on 15 years of experience with the language.

This book has been designed to facilitate the learning of COBOL. It includes thorough coverage of concepts, as well as examples of applications at both elementary and intermediate levels of complexity. Review sections are included in the text as opportunities for the reader to enhance effective learning through self-testing. Students in business administration and computer science will find this book to be a suitable guide to the language.

Throughout this book, the concepts of structured programming are used. A structured program is one which follows certain principles of modular design and results in simplification of programming tasks and improved self-documentation. The COBOL language is particularly well-suited for the application of structured programming concepts, and use of these concepts has been incorporated in all of the computer programs that are described.

The book consists of 16 chapters. The first three chapters present a complete set of basic concepts, so that the student can begin to write complete programs in a relatively short time. Chapter 4 then presents concepts and methods of program design and lays a foundation for the structured programming approach employed throughout the text.

Beyond the introductory material of the first three chapters, the language concepts and options are developed at three levels. Chapters 4 and 5 present a basic subset of DATA and PROCEDURE statements. Chapter 6, 7, and 8 present more advanced features that enable the student to master the language at an intermediate level. Finally, Chapters 9 through 16 develop special language features as they relate to the general application areas of sequential file processing, sorting and merging, table handling, indexed sequential and relative file processing, subroutine programming, and the report writer feature.

As compared with the previous edition of this book, the application of structured programming has been expanded to include new concepts in program design, as well as program structure in general. Several chapters have been restructured so that students will be able to write intermediate-level programs sooner. New exercises have been added throughout the book, file processing has been extended to three separate chapters, and new coverage is included in the two new chapters on program structure and design and on the report writer feature.

The authors express their appreciation to Charles E. Stewart for his very capable supervision of this project. We also extend thanks to several anonymous reviewers for their comments and recommendations with respect to this revision.

A. S. Philippakis
Leonard J. Kazmier

ACKNOWLEDGMENT

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

> Any organization interested in reproducing the COBOL standard and specifications in whole or in part, using ideas from this document 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 the following acknowledgment paragraphs in their entirety as part of the preface to any such publication (any organization using a short passage from this document, such as in a book review, is requested to mention 'COBOL' in acknowledgment of the source, but need not quote the acknowledgment):
>
> 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 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.

CONTENTS

1 Programming for administrative data processing

WHAT IS A DATA PROCESSING SYSTEM?

A well-designed data processing system provides managers with information needed for decision making by providing for efficient data collection, data processing, and dissemination of the resulting information. Such a system can include manual procedures, the use of business forms, electronic data processing equipment, data files, and computer programs. Administrative data processing has a number of distinct features with respect to data input, the processing of data, and the output of such systems.

Data *input* typically is voluminous and consists of numeric and nonnumeric data records of varying length. For example, a sales data record generally includes the name of the customer, the date, the amount of the purchase, the item description, and the like. In a retail business, these sales data records easily can total thousands per week.

The *processing* of business data input is characterized by the updating of files. Business activities require large files of data in order for the data processing system to be a fairly accurate reflection of these activities. Data files constitute a symbolic model of the activities taking place in an organization. Keeping this model current is the function of *file updating.* Input data are processed with respect to historical files in order to update the data in the files. For example, a bank maintains a file of its checking account customers that reflects the state of the bank's checking account function. When input data are accumulated in groups and processed periodically, we have a *batch processing* system, and the data files reflect the state of the activity at a point in time. When input data are processed as each transaction occurs, then the files reflect the current state of business activity, and we have a *real-time* system. File updating requires complex logic because files contain a variety of data and the processing logic must foresee all possibilities that can arise. For instance, a payroll system is a data processing task that often requires many months of effort, even though at first glance it seems a trivial matter involving simple arithmetic.

The *output* of administrative data processing is characterized by the production of reports that group and summarize data by meaningful categories which correspond to various functions.

In contrast to administrative data processing, scientific computing typically is characterized by a relatively low volume of input, small or nonexistent files, less complex processing logic but extensive arithmetic manipulations, and more limited report production needs.

Since administrative data processing has characteristics different from those of scientific computing, a special language has been developed to accommodate the particular needs associated with such processing of data. The language is COBOL (COmmon Business Oriented Language). The counterpart languages for scientific computation are FORTRAN (FORmula TRANslation) and ALGOL (ALGOrithmic Language). PL/I (Programming Language I) was designed for both administrative and scientific applications but has not been used widely. Although COBOL is the most widely used language for administrative data processing in conjunction with both small and large computers, two other languages also frequently used for data processing with small computers are BASIC (Beginner's All-purpose Symbolic Instruction Code) and RPG II (Report Program Generator II).

COMPUTER SYSTEMS FOR DATA PROCESSING

A system is a set of interrelated parts. Computers are referred to aptly as computer systems, since they consist of interrelated component parts. Figure

1-1 shows the basic component parts of a computer system: input, central storage, auxiliary storage, central processing, and output.

The input component performs the function of transmitting data from an external storage medium to the central storage of the computer. One well-known input device is the card reader, which transmits data from punched cards to central storage. The recording of data on punched cards is performed offline using a keypunch. Because punched cards continue to be used very widely, and because the chances are that the student of this text will use a keypunch for data and program preparation, the appendix at the end of Chapter 2 provides instruction on the use of the keypunch.

A variety of other input devices exists in addition to the card reader. These include magnetic tape, magnetic disk (often created by key-to-tape or key-to-disk devices), keyboard terminals, cathode-ray-tube (CRT) terminals, electronic "cash registers," paper tape, optical readers, and magnetic ink readers. Several types of input-output devices are presented in Figure 1-2.

Central storage is used to hold program instructions and data being processed by a given program. Central storage is measured in terms of *bytes* or *words*, depending on the hardware architecture of a computer system. Each byte or word is addressable by the central processor and can be made available for input or output.

Because central storage is relatively expensive and limited, business computer systems include auxiliary or secondary storage. The main use of auxiliary storage is to hold the data files of the data processing system. Magnetic tapes and disks are the commonly used devices, but magnetic

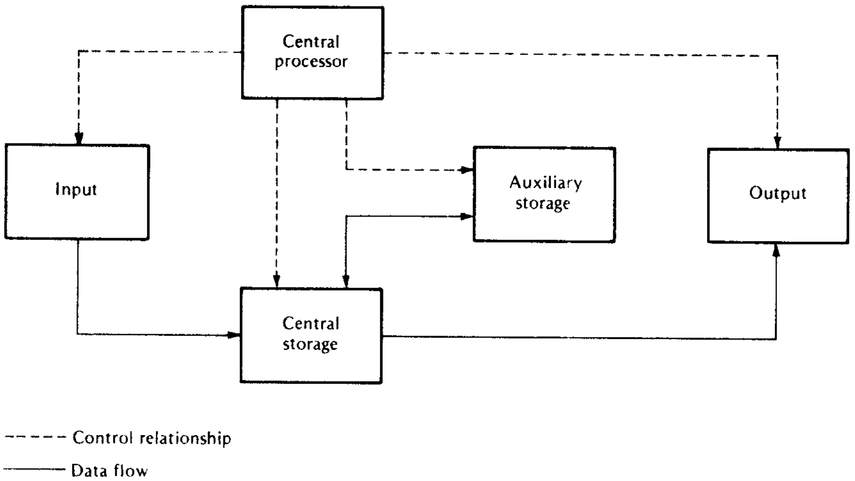


FIGURE 1-1 COMPONENTS OF A COMPUTER SYSTEM.

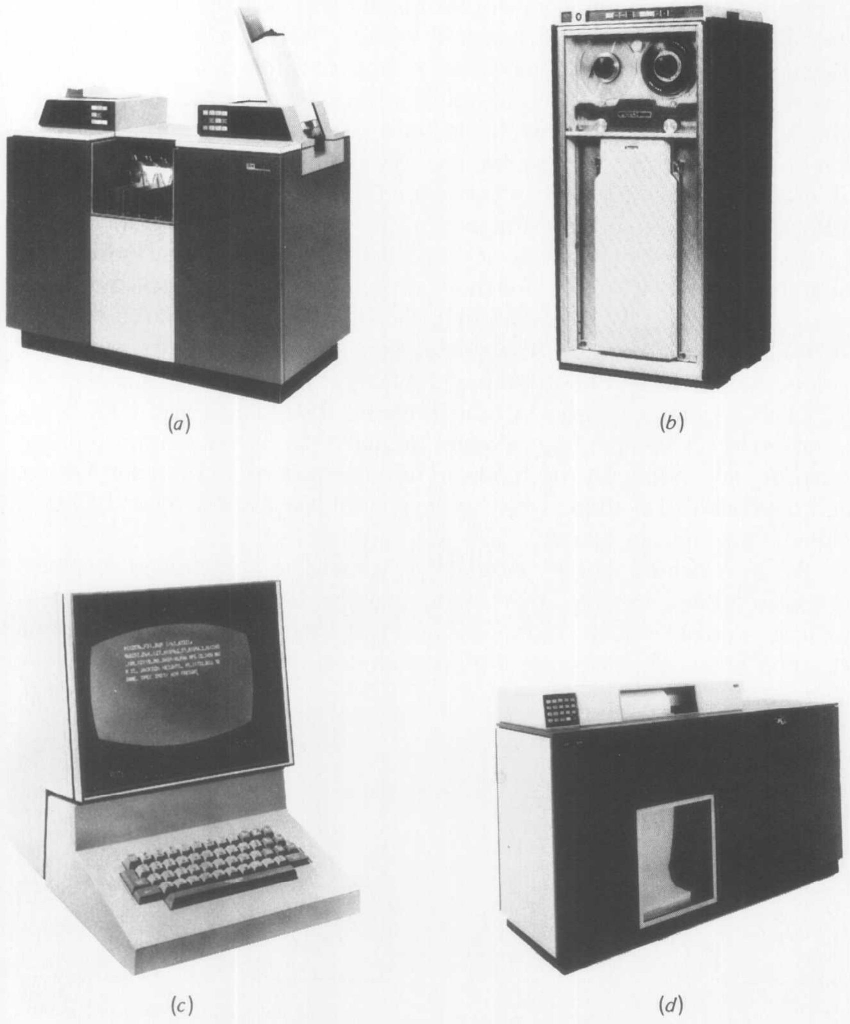
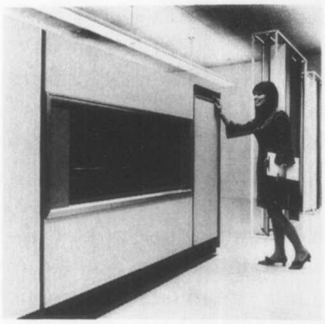
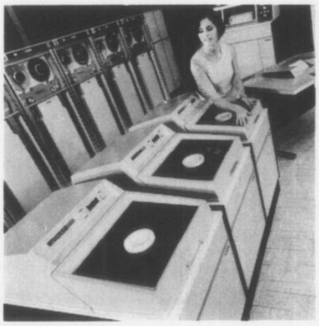


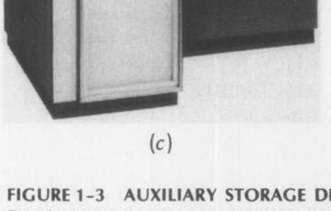
FIGURE 1-2 INPUT/OUTPUT DEVICES. (*a*) **IBM 2540 CARD READER/CARD PUNCH** (IBM CORPORATION). (*b*) **UNIVAC MAGNETIC TAPE UNIT** (SPERRY RAND CORPORATION). (*c*) **IBM CATHODE-RAY TUBE** (IBM CORPORATION). (*d*) **IBM 3211 PRINTER** (IBM CORPORATION).



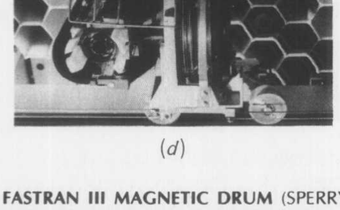
(*a*)



(*b*)



(*c*)



(*d*)

FIGURE 1-3 AUXILIARY STORAGE DEVICES. (*a*) **UNIVAC FASTRAN III MAGNETIC DRUM** (SPERRY RAND CORPORATION) (*b*) **UNIVAC 8414 MAGNETIC DISK STORAGE** (SPERRY RAND CORPORATION). (*c*) **IBM 3321 DATA CELL UNIT** (IBM CORPORATION). (*d*) **IBM 3850 CARTRIDGE SYSTEM** (IBM CORPORATION).

drums and extended cartridge storage systems also are used. A reel of magnetic tape when mounted on a tape drive becomes *online* and represents auxiliary storage. A reel of tape on a storage shelf is *offline* and does not constitute auxiliary storage as a component part of a computer system. Similar considerations apply to removable disks. Several types of auxiliary storage devices are presented in Figure 1-3.

A frequently used output device is the printer, which records data from central storage in printed form. In administrative applications, printed reports are common and voluminous. Recent developments have resulted in printer speeds approaching 20,000 lines of output per minute. Magnetic tapes and disks are also usable for output, as are punched cards, CRT display units, and microfilm.

The central processor controls and coordinates the functions of the other components of a computer system and includes the arithmetic and logical capabilities that characterize electronic computers.

It should be noted that most computers are *general purpose* digital computers, and a given computer system may be used equally well for administrative data processing or for scientific computing; however, the specific choice of components for a computer system is influenced by the nature of the expected application. A computer system used by a research laboratory consists of a mix of specific components different from that of a system used by an insurance company. The input, output, and file storage requirements are quite different.

REVIEW

1 The entry of data into a computer system is called data ______.

input

2 After data are input, the performance of such functions as file updating is included in the category of activities called data __________.

processing

3 When data are accumulated and processed periodically, ______ processing is involved. When input data are processed as each transaction occurs, a(n) __________ system is involved.

batch; real-time

4 Such activities as the production of printed reports and the display of information by means of a CRT are examples of computer __________.

output