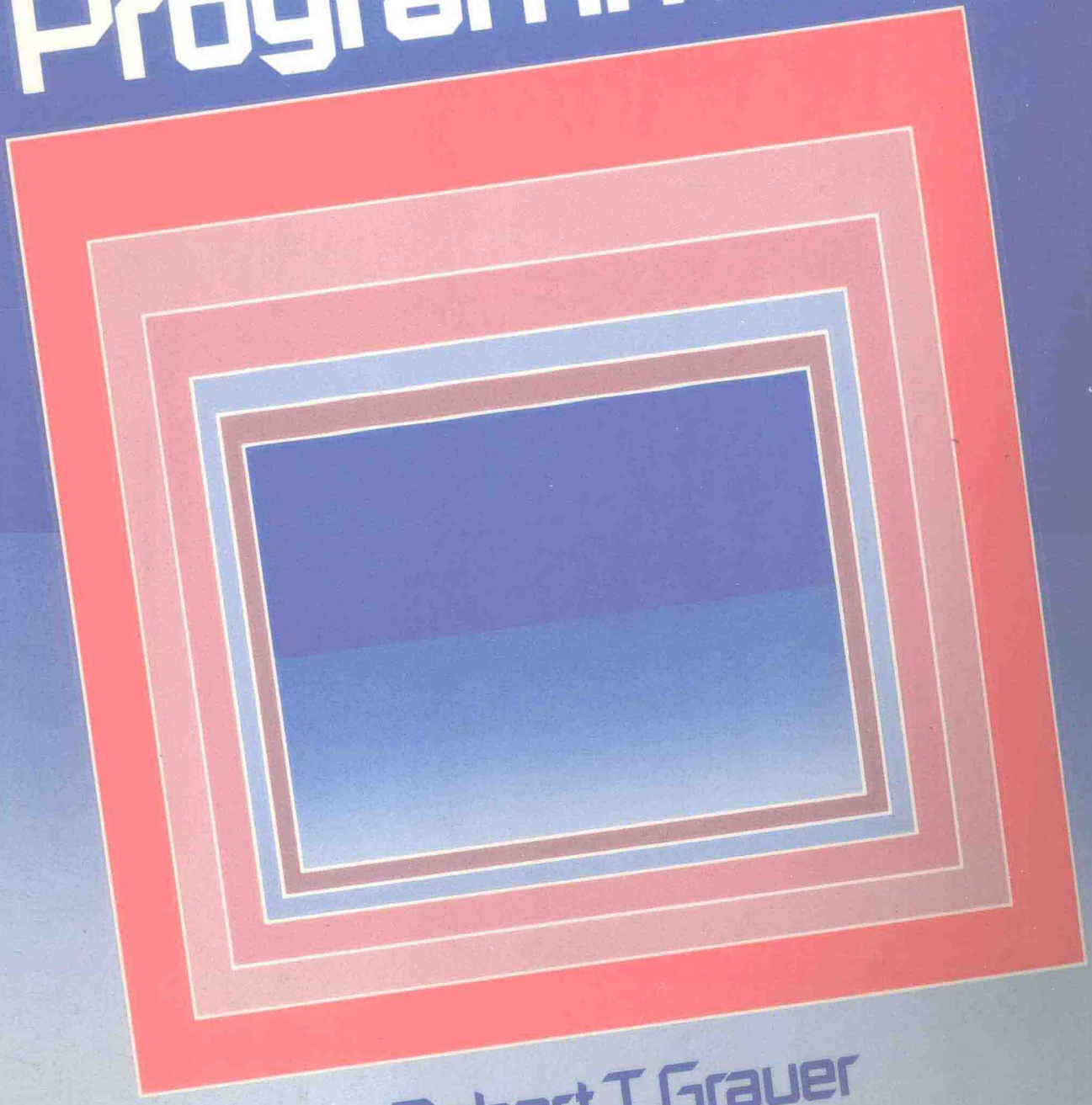
Structured COBOL Programming



Robert T. Grauer

Structured COBOL Programming

Robert T. Grauer, Ph. D.
University of Miami, Florida

PRENTICE-HALL, INC., Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data
Grauer, Robert T. 1985
Structured COBOL programming,
Includes index.
1. COBOL (Computer program language) 2. Structured programming. I. Title.
QA76.73.C25G7373 1985 001.64'24 84-26358
ISBN 0-13-854217-1

Editorial/production supervision: Lynn S. Frankel
Interior and cover design: Anne T. Bonanno
Manufacturing buyer: Gordon Osbourne

Printed in the United States of America

10 9 8

ISBN 0-13-854217-1 01

Prentice-Hall International, Inc., London
Prentice-Hall of Australia Pty. Limited, Sydney
Editora Prentice-Hall do Brasil, Ltda., Rio de Janeiro
Prentice-Hall Canada Inc., Toronto
Prentice-Hall Hispanoamericana, S. A., Mexico
Prentice-Hall of India Private Limited, New Delhi
Prentice-Hall of Japan, Inc., Tokyo
Prentice-Hall of Southeast Asia Pte. Ltd., Singapore
Whitehall Books Limited, Wellington, New Zealand

Structured COBOL Programming

To Marion, Benjy, and Jessica

Preface

Structured COBOL Programming is a truly comprehensive work, providing in a single source all subjects normally covered in the one-year COBOL sequence. The scope is extensive, ranging from an introduction to COBOL to maintaining sequential and nonsequential files. The single-volume presentation affords a unified approach and eliminates the need for separate texts in the two semesters. This is especially valuable when students transfer schools (e.g., from a community college) for the second semester, as it eliminates the problem of what may or may not have been previously covered.

Structured COBOL Programming represents several years of effort, encompassing many revisions to earlier books by the same author. It embodies suggestions from hundreds of individuals, both students and instructors, who have used one Grauer book or another. Many new features have been added, while others have been retained, with the following of special interest:

- *Immediate entry into COBOL programming,* beginning in Chapter 1. Programming is learned by doing, and the book has the student writing a complete program from the very beginning. Chapter 2 continues the discussion, by having the reader "punch and run" the program of Chapter 1 in a thorough introduction to the programming process.
- *Twenty-Two Complete COBOL Programs,* which reinforce the discussion in the text and serve as both pedagogical aids and subsequent reference material. Each illustrative program is presented in a uniform and detailed format including program narrative, record layouts, report layouts, test data, and processing specifications.
- *Increased Number of Programming Assignments,* which appear at the end of each chapter. The assignments are presented in the same format as the illustrative programs.
- *Abundant and Challenging Exercises,* at the end of each chapter. The number of exercises has been increased from earlier books, and the problems are constructed to reinforce and stimulate the reader's ability in COBOL.
- *New Debugging Workshops,* featuring an additional eight COBOL programs to give students practice in an area where it is needed most.
- *Ten Programming Tips,* which go beyond the syntactical rules of COBOL and suggest stylistic considerations to make programs easier to read. Many of the tips take the form of coding standards and are typical of what the student will encounter in "the real world."
- *Extensive Use of Graphic Aids,* including a two-color presentation and abundant use of tints. This enables important elements, such as COBOL syntax and 8X changes and enhancements, to be consistently highlighted from chapter to chapter.

- *New System Concept Presentations,* at the beginning of each of the advanced COBOL chapters. Instruction in COBOL has come to require additional material, beyond the language itself. These discussions include control breaks, data validation, logical versus physical records, techniques for table lookups and initialization, sorting, Report Writer, the balance line algorithm for file maintenance, and organization of indexed files.
- *Thorough presentation of structured methodology,* with emphasis on hierarchy charts and pseudocode, and deemphasis of the traditional flowchart. Material is also included on stepwise refinement, top-down testing, and Nassi–Shneiderman diagrams.
- *Substantial coverage of the 8X compiler,* which is both thorough yet nonconfusing to those familiar with COBOL 74.

APPROACH TO THE NEW COMPILER

The text is designed for users of both the 74 compiler and the new 8X version. All listings were run under COBOL 74 and are upwardly compatible. In addition each chapter contains a concluding section that presents the relevant 8X changes and enhancements. Appendices A, B, and C contain still more COBOL 8X material, including a sample program. Appendix D contains the 8X Language Summary.

The moderate approach to the new compiler was carefully chosen. It would have been possible, for example, to inundate the reader with new features of COBOL 8X to the point of making the illustrative programs totally incompatible with COBOL 74. To have done so would have greatly weakened the book's overall impact. It took many years for industry to convert from COBOL 68 to COBOL 74, and history has a tendency to repeat itself.

ACKNOWLEDGMENTS

The author wishes to thank the many people who have made this book possible. He acknowledges the contributions his students have made over the years through their many suggestions and active class participation. He is indebted to the in-depth analysis provided by his reviewers, including: Professor Earl Adams (Illinois Central College), Professor Carol C. Grimm (Palm Beach Community College), Professor Margot Hummer (University of Miami), Professor Lewis Myers, Professor James Payne (Kellog Community College), Professor Joel Stutz (University of Miami), Professor Terry Walker (University of Southwestern Louisiana), and Mr. Richard Weiland.

Cathy Benway, Jackie Clark, and Lita Fox, all of the University of Miami, helped to proofread the book. Each made a significant contribution. Ms. Sheila Grossman typed, then retyped the manuscript, always maintaining her sense of humor.

Jim Fegen and Marcia Horton are the acquisition editors responsible for this project. Lynn Frankel, the production editor, put all pieces together in her expert way. Anne Bonanno designed the book and its cover. Herb Daehnke did his usual expert job on the computer listings. Karen Fortgang put on the finishing touches. Last, and certainly not least, thanks to Joe Heider, marketing manager, for making the book a success.

A final word of thanks to the hundreds of unnamed students in MAS223 at the University of Miami who made it all worthwhile.

ROBERT T. GRAUER

Contents

Introduction

Overview

This book is about computer programming. In particular it is about COBOL, a widely used commercial programming language. Programming involves the translation of a precise means of problem solution into a form the computer can understand. Programming is necessary because, despite reports to the contrary, computers cannot think for themselves. Instead they do exactly *what they have been instructed to do, and these instructions take the form of a computer program. The advantage of the computer stems from its speed and accuracy. It does not do anything that a human being could not do, given sufficient time.*

All computer applications consist of three phases: input, processing, *and* output. *Information enters the computer, it is processed (i.e., calculations are performed), and the results are communicated to the user. Input can come from punched cards, a diskette, magnetic tape, a hard disk, computer terminals, or any of a variety of other devices. Processing encompasses the logic to solve a problem, but in actuality all a computer does is add, subtract, multiply, divide, or compare. All logic stems from these basic operations, and the power of the computer comes from its ability to alter a sequence of operations on the basis of the results of a comparison. Output can take several forms. It may consist of the ubiquitous 11 × 14⅞ computer listing or printout, or it may be payroll checks, computer letters, mailing labels, magnetic tape, etc.*

We begin our study of computer programming by describing a single problem and then developing the logic and COBOL program to solve it. This rapid entrance into COBOL is somewhat different from the approach followed by most textbooks, but we believe in learning by doing. *There is nothing very mysterious about COBOL programming, so let's get started.* □

THE FIRST PROBLEM

Our first problem involves a set of student records, one record per student. Each record contains the student's name, number of completed credits, and the student's major.

We are to produce *a list of engineering students who have completed at least 110 credits*. Our problem is to develop a COBOL program that can process the set of student records to generate the desired report.