

ADVANCED STRUCTURED COBOL

Second
Edition

Mike Murach
John Padgett



Advanced Structured COBOL

Second Edition

Mike Murach

Mike Murach and Associates

John Padgett

University of Southern Colorado



SCIENCE RESEARCH ASSOCIATES, INC.
Chicago, Henley-on-Thames, Sydney, Toronto

An IBM Company

Acquisition Editor	Michael Carrigg
Project Editor	Richard E. Myers
Copyediting	Timothy Loughman
Text and Cover Design	Harry Voigt
Production Administrator	Timothy Taylor
Composition	Interface Studio

Acknowledgments

Cover photograph: Copyright ©1985, L. R. Lewandowski.
All rights reserved. Reproduced by permission of the photographer.

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 COBOL 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 materials used herein, FLOW-MATIC (trademark of Sperry Rand Corporation), Programming for the UNIVAC (R) 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.

Acknowledgments continued on page 525.

Library of Congress Cataloging-in-Publication Data

Murach, Mike.

Advanced structured COBOL.

Includes index.

1. COBOL (Computer program language)
2. Structured programming. I. Padgett, John, 1936-
II. Title.

QA76.73.C25M858 1987 005.13'3 86-31660
ISBN 0-574-21990-0

Copyright © Mike Murach and Associates, 1987.

All rights reserved.

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

Advanced Structured COBOL: Second Edition

Preface for Instructors

An advanced COBOL course should concentrate on bringing students programming skills as close to professional level practice as possible in the classroom. With that goal in mind we have organized this text to help ensure that the student is reminded of the foundations learned in the beginning COBOL course, that structured programming concepts are reviewed and enhanced, that reporting techniques are strengthened, and that file concepts are thoroughly covered.

As you review the table of contents for this text, you will see that the bulk of the pages are devoted to file concepts and file handling. We feel that one of the most challenging topics for students is that of file update. When they have successfully completed update and other file-related programs, they will have significantly improved their programming logic skills, they will have enhanced their knowledge of file storage, and they will have tested their ability to handle large and multi-faceted programs.

Our experience with teaching Advanced COBOL students for several years leads us to believe that the most important thing that students can learn in this course is what we call “ways of work.” The concept of ways of work includes the preparatory activities—structure chart, pseudocode, and test plan—that should be completed before beginning to code a program. Ways of work includes using a carefully constructed test plan to guide a phased development of the program. Ways of work includes coding and testing the program in manageable chunks instead of trying to code and test it all at once. Ways of work includes several techniques available to aid in debugging programs and verifying their accuracy.

Before going on to the next subject, you might wish to read the Preface for Students. The individual chapters, their role in the course, and possible variations in the order of study are discussed there.

The Goals of Advanced Structured COBOL and How They Are Met

Advanced COBOL, as stated, should enable the successful student to step from the classroom into a position as a programmer trainee, with sufficient skills to become productive after a short period of becoming acclimated to the particular environment in which he or she has chosen to work. In order for a text significantly to help the student reach that level, it must have some specific goals and means for achieving them. The following lists the goals of

this text and the topics, discussions, and projects that make possible the attainment of the goals.

1. The student learns advanced COBOL facilities and techniques.
COPY, subprograms, and tables with indexes are reviewed.
The character manipulation statements INSPECT, STRING, and UNSTRING are covered thoroughly.
Report Writer is introduced.
The COBOL statements for specifying and accessing sequential, indexed and relative files are presented.
The problem set contains programs dealing with tables, the COPY statement and subroutines. Other problems involve the use of the character manipulation statements. Problems for practicing Report Writer and for file handling are included.
2. The student learns the essential record and file concepts.
The types of files are discussed.
Guidelines for designing master file records are presented.
The advantages and disadvantages of the different file organizations are presented.
Variable-length records and the COBOL statements for handling them are presented.
The different requirements for sequential, indexed and relative files are thoroughly discussed.
How indexed-sequential files are stored and accessed is thoroughly discussed.
Multi-indexed file concepts are presented.
3. The student learns important file-related concepts and methods.
Transaction editing and its importance are presented.
Program problems involving transaction editing are part of the problem set.
Transaction input and editing over the video display terminal are thoroughly discussed.
A problem and problem ideas for interactive transaction programming are given in the problem set.
4. The student learns to develop large programs quickly, efficiently and accurately by doing the development in stages.
Top-down coding and testing are reviewed.
Large program problems are presented in stages.
5. The student learns to program the logic of file update and random file access.
Three versions of sequential file matching logic are contrasted and thoroughly discussed.
Programming examples for sequential, indexed and relative file update are thoroughly discussed.
Comprehensive program problems covering sequential, indexed and relative file update have been developed for student practice in the problem set.
A multi-indexed retrieval program problem is part of the problem set.

6. The student becomes aware of the evolution of COBOL, of the latest ANSI standards, and of current theory about programming.

The text presents the important aspects of the ANSI 1985 standards.

Structured concepts, including cohesion and coupling, are discussed.

Structured style and program organization is reviewed.

The COBOL statement formats in the appendix conform to the ANSI 1985 standards.

The Genesis of Advanced Structured COBOL

Mike Murach and Associates, the original authoring group, developed the first edition. The first edition was one of the few COBOL textbooks that presented and supported structured programming in a practical, understandable manner. The team that developed the first edition employed their collective high level of professional expertise and long experience in presenting a very successful COBOL text.

When the time arrived for the first edition to be revised and modernized, Mike Murach was unable to do the revision because of other business interests. The publisher, Science Research Associates, contacted me to see if I would be interested in doing the revision. As an Associate Professor of Computer Science Technology, and a former professional programmer and analyst myself, I had been using the first edition of Murach's *Structured COBOL* for many semesters. I gladly accepted the opportunity to continue the life of a text that I considered to be one of the finest available for teaching structured COBOL.

The second edition has been enhanced with more detailed program problems, with fuller explanations and guidelines for structured programming and style, with reference to the ANSI 1985 standards, with new topics such as interactive terminal processing, and with thorough explanations and examples of file processing and update logic. Where the first edition contained just enough material to cover a two-semester COBOL course, the second has been split into two textbooks to cover the fundamentals course and the advanced course. The expanded coverage gives instructors more flexibility in dealing with topics they may wish to emphasize.

In preparing the second edition, I have tried to remain faithful to the fine work begun in the first edition while enhancing that work in both breadth and depth.

Declaration of Appreciation

As coauthor, I recognize that developing a successful textbook requires a strong team effort, and I would like to express by appreciation to the following people for their parts in that development.

- First I am grateful to Mike Murach and Associates for their work which resulted in the very successful first edition of *Structured COBOL*.
- Secondly the team that SRA, the publisher, brought together should ensure the success of this edition. Specifically, I need to thank Timothy Loughman, the copyeditor for the many hours he spent with me in revising and polishing the text itself. Michael Carrigg, the acquisition editor,

and Richard Myers, SRA's manager of editorial development, deserve high praise for overseeing the development and helping maintain a tight production schedule. Bernie Esposito, University of Baltimore, Eli Boyd Cohen, California State College-Sacramento, and Bill Knouth, Heald College, reviewed the first draft of *Advanced Structured COBOL*. Each of these reviewers provided helpful comments and suggestions.

■ Next I want to express my appreciation to my colleagues and family. My colleagues at the University of Southern Colorado have been very supportive of my efforts, and have often picked up the slack left when my work on the book took away from time available to counsel students or to serve on committees. My family has been equally supportive and understanding when I gave them less attention than they deserve.

■ Finally, our local programmer and technical reviewer, Molly Jagger, is one to whom I cannot adequately express my appreciation. Mrs. Jagger has written several program examples and prepared sample program solutions for the instructor's guide. She also reviewed the entire text for technical problems, and her effort has minimized the errors, oversights, and incomplete explanations that naturally occur in an effort of this magnitude.

John J. Padgett
Pueblo, Colorado

Preface for Students

Having successfully completed the fundamental COBOL course, you have developed basic knowledge about and skill in using the COBOL programming language. Now you are ready to extend that knowledge and skill to the level where you can present your application and yourself to an employer with confidence. When you successfully complete the advanced COBOL course, you will need only experience and maturity to become an expert programmer.

Before beginning the advanced course there are a few aspects of the course and this text about which you should be aware:

1. The coding and testing of the problem programs in this text require a more phased approach than you may be used to. The programs, particularly those for file update, are complex in that they combine several distinct activities. We cannot emphasize enough the importance of developing your programs “by the numbers,” as the military would say. That means that you should carefully develop a structure chart for the general program design, reworking it as many times as necessary until you feel that it is as good as you can make it at that stage. After that, you should either do the pseudocode for each module, or develop a phased test plan and do the pseudocode for the modules in the first phase(s) of the plan. The actual coding and testing should proceed in phases as indicated by the test plan. Do not try to code and test the programs in this text (especially those for file update) all at once. Rather you should code, test and revise one phase until it works properly before moving on to the coding and testing of the next phase. This is why the concepts and practice of top-down coding and testing are so important.

2. The format of each topic is roughly the same. The topic is introduced and is followed by a list of new terms that will be encountered in the topic, and the objectives for the topic. Then the subject discussion follows and the topic typically ends with a summarizing discussion and with problems related to the topic and with their attendant solutions.

You should be sure that you are comfortable with the terms, since knowing them will help your understanding of the subject matter discussions. The terms are limited to those that should be new to you. The objectives for the topic follow the terminology list. These are

usually stated in terms of behavioral objectives—in other words, things that you should be able to do when the topic is successfully completed. The objectives can usually be achieved only by actually writing a program to develop the skills that you are expected to acquire.

You should study the end-of-topic problems and solutions carefully as they are intended to lend insight into the subject matter and its application. They also provide an introduction to the problem programs you will be asked to implement.

We recommend that you begin to study the problem program that will be assigned for each chapter or topic as soon as the subject is introduced. You should be continually thinking how to apply the new learning to the problem as the subject is discussed.

3. This text is designed so that the chapters do not have to be covered in sequence. Since you have developed basic COBOL knowledge and skill in your fundamentals course, the topics and problems in this text may be selected according to the particular goals of your advanced course. We would like briefly to review the chapters of this book:

- Chapter 1 is intended to be a review. We suggest that you read through it and carefully study any area where you feel a weakness and bypass those areas where you feel adequately prepared. For example, if you covered the COPY statement and subprograms thoroughly in your fundamentals course, you can go on to apply that knowledge in the programs that require the COPY statement and subprograms.

Of special importance is the topic of top-down coding and testing. We feel that this topic should be covered immediately, even if you are familiar with the procedure described. Many of the problem programs in this book will require the use of top-down coding methods in order to develop the programs in the most efficient and accurate manner possible. Failure to practice those methods will prolong the development time longer than necessary, and cause you needless frustration.

- Chapter 2 covers structured concepts. We feel that the style and organization guidelines presented are among the best, but you may have developed your own set with which you are satisfied. If so, you may only need to skim through the guidelines presented here for contrast, comparison, and possible new ideas. The topics on coupling and cohesion, and the ANSI standards could be covered toward the end of the semester when there is not time to introduce new programming concepts but there is time to discuss subjects of more general interest to the professional programmer.

- Chapter 3 treats character manipulation, transaction edits, and validation practices. Again, you may wish to bypass those topics already covered in your fundamental course. Or, the topics may be covered toward the end of the course, when it will be time to integrate the concepts of validate/edit, sort, and update for batch update or validate/edit and update for interactive update.

- Chapter 4 discusses reporting, and covers design and the COBOL facility Report Writer. If you do not have Report Writer available, then that topic could be bypassed or just studied briefly. Report design may be covered any place in the course, because no programming skills depend upon a knowledge of that topic.

- Chapter 5 is a necessary chapter. The first two topics on record design and variable-length records may be covered at other times, but the third topic, dealing with update master records, should be studied prior to beginning Part III.
- Part III, which treats file handling, is the focal point of the text and probably should be studied in sequence as it moves from sequential organization (the basic file structure) to indexed and relative organizations (the basic random structures).
- Chapter 6 is probably the most important chapter in the text. Here is where you begin to understand how records are stored on files, especially random-access files.
- Chapter 7 covers sequential file update logic. The problems associated with this chapter will provide you with the greatest challenge of your COBOL programming experience. The challenge will come not from the programs being intentionally difficult, but from the fact that the update process is many-faceted and the programs are naturally longer.
- Chapter 8 is about indexed files and its topics are also very important. In it you study the currently most popular random file organization method for applications programs.
- Chapter 9 covers relative files. It should be studied so that you can begin to understand how to use relative files, but you may not find time to write programs using that organization. If such is the case, it will be all right—the most often-used organizations for COBOL applications are sequential and indexed, and that is where you should concentrate your learning.
- Part IV, which consists of chapter 10, may be studied at any time. Knowing how to program interactive applications that use terminals is a skill that more students should develop, and we recommend that it be studied and used in at least one program problem.
- The appendixes contain the problem set for the course and the COBOL reference formats. There are more problems here than you could be expected to complete in one course, so they should be selected according to your skill level when you begin this course and to the goals of the course. The problem specifications are typically very detailed so as to minimize the confusion and ambiguity that can occur with less specific problem statements. In addition, we feel that you will be confronted with similarly detailed specifications (or the need for them) as a professional programmer.

The reference formats have been made to conform to the 1985 ANSI standard format specifications. Some of the formats will not be available on your computer due to the level of compiler you use. Some will not yet be available because the standards were only finalized in 1985 and it takes time to incorporate them into a compiler.

This book provides you with coverage of subjects that are designed to enable you to enter into the professional world with a sound foundation of knowledge and skill and with confidence in your ability to succeed. To achieve that end you will have to avoid waiting until the last minute to begin a program, and you will have to employ the phased approach to program development. We know that, when you combine these two practices with diligent

daily effort, you will amaze yourself with what you can accomplish.

We hope that you find this book stimulating to your professional growth and exciting to read because of the new insights into professional programming you will gain. Let us now begin.

John J. Padgett
Pueblo, Colorado
1987

Contents

Preface	xi
Preface for Instructors	xi
Preface for Students	xv

Part One: Introductory Concepts

Chapter 1	Review	3
	Topic 1 Top-down Coding and Testing	3
	Orientation—Terminology and Objectives	3
	Coding and Testing	4
	When to Use Top-Down Testing	4
	Approaches to Top-Down Testing	5
	How to Write a Top-Down Coded Program	6
	How to Code Programming Stubs	7
	Discussion, Problem and Solution	8
	Topic 2 Tables with Indexes	11
	Orientation—Terminology and Objectives	11
	Table Concepts	11
	Table Use	11
	Table Organization	11
	Table Access	11
	Specifying, Loading and Accessing Tables	12
	Discussion, Problems and Solutions	18
	Topic 3 The COPY Facility	23
	Orientation—Terminology and Objectives	23
	The COPY Statement	24
	When to Use the COPY	26
	Discussion, Problems and Solutions	26
	Topic 4 Subprograms	28
	Orientation—Terminology and Objectives	28
	Statements for Subprograms	28
	Discussion, Problems and Solutions	29

Topic 5 Issue: Use of COPY vs. Subprogram	32
 Chapter 2 Structured Concepts	35
Topic 1 Structured Style and Program Organization	35
Orientation—Terminology and Objectives	35
Review of Structured Style Guidelines	36
Basic Style Guidelines	36
Other Important Style Guidelines	38
General Style Guidelines	38
Identification Division Guidelines	39
Environment Division Guidelines	39
Data Division Guidelines	39
Procedure Division Guidelines	41
Discussion, Problem and Solution	46
Topic 2 Cohesion and Coupling	53
Orientation—Terminology and Objectives	53
Cohesion	54
Minimizing Data and Temporal Cohesion	55
Eliminating the Other Non-functional Cohesion Types	56
Clues to Creating Modules with Functional Cohesion	56
Coupling	56
Creating Loosely Coupled Modules in COBOL	58
Discussion	58
Topic 3 ISSUE: ANSI COBOL 1985	59
Orientation—Terminology and Objectives	59
Significant Enhancements to ANSI COBOL	59
The END— . . . Statements (Scope Delimiters)	60
Other PERFORM Enhancements	61
EVALUATE—The Case Construct	62
The FALSE Condition Tests	62
User-defined Data Sets	63
Enhancements to the SET Statement	63
Other Significant 1985 Standards	64
Discussion	64
 Chapter 3 Character Manipulation	65
Topic 1 COBOL Character Manipulation Commands	65
Orientation—Terminology and Objectives	65
Uses of the Character Manipulation Commands	66
The INSPECT Command	67
The STRING Command	70
The UNSTRING Command	72
Discussion, Problems and Solutions	75

Topic 2 Editing—Batch and Interactive Modes	77
Orientation—Terminology and Objectives	77
The Editing Environment	77
Batch Edit with Card Input	78
Batch Edit with Terminal Input	79
Interactive Edit and Update	79
The Types of Editing	80
Editing Actions	83
Discussion, Problems and Solutions	84

Part Two: Reporting and Master Records

Chapter 4 Reporting	93
Topic 1 Report Design	93
Orientation—Terminology and Objectives	93
Designing Reports	94
Report Types	94
Report Line Types	96
Headings	96
Detail Lines	96
Footers	96
How Data Fields Should Look	97
Edit Symbols	97
Total Amounts	98
Column Headings	98
How the Total Page Should Look	99
White Space	100
Field Relationships	101
Report Filing/Use Considerations	102
Top Binding	102
Side Binding	102
Standard Size Paper	102
Discussion, Problems and Solutions	104
Topic 2 The Report Writer Feature	110
Orientation—Terminology and Objectives	110
Using Report Writer	111
Data Division Elements	112
Procedure Division Elements	124
Discussion, Problem and Solution	125
 Chapter 5 The Master File Record	 129
Topic 1 Designing the Master Record	129
Orientation—Terminology and Objectives	129
Record Design	130
Determining Field Specifications	130
Organizing the Fields Within the Record	131

Variable-length Records and Multiple Record Formats	133
Variable-Length Records	134
Multiple-Record Formats	137
Determining What Fields Should Be in a Record	137
Discussion, Problems and Solutions	140
Topic 2 COBOL Commands for Variable-Length Records	142
Orientation—Terminology and Objectives	142
Specifying a Variable-Length Record	143
Processing a Variable-Length Record	146
Discussion, Problems and Solutions	154
Topic 3 Updating the Master File Record	159
Orientation—Terminology and Objectives	159
The Types of Update Records	160
Formats of Update Records	160
Updating the Master Record	166
Discussion, Problems and Solutions	173

Part Three: File Handling Concepts

Chapter 6 File Concepts	179
Topic 1 File Hardware Concepts	179
Orientation—Terminology and Objectives	179
How Data Is Recorded	180
Hardware Types	182
Discussion, Problems and Solutions	190
Topic 2 Direct Access Storage Concepts	190
Orientation—Terminology and Objectives	190
File Organization	191
An Inventory File Example	191
A Sequential File Example	194
A Directly Addressed File Example	195
An Indirectly Addressed File Example	197
An Indexed Sequential File Example	202
Adding Records to an Indexed File	206
Discussion, Problems and Solutions	209
Topic 3 Updating the Master File	215
Orientation—Terminology and Objectives	215
The Update Process	215
Updating a Sequential File—Simple Update	217
Updating a Sequential File—Complex Update	219
Updating a Random File	233
Discussion	241
Topic 4 COBOL Statements for Organization and Access	241
Orientation—Terminology and Objectives	241
Specifying the File Organization	242

Specifying the File Access Mode	242
Procedure Division Commands	243
The Basic Input/Output Commands	245
Additional Input/Output Commands	248
Discussion	250
Chapter 7 Sequential Files—Matching Logic	253
Topic 1 Simple Sequential Matching Logic and Proving	253
Orientation—Terminology and Objectives	253
Simple Update Record Matching Logic	255
A Tabular Method for Proving Logic	259
Selecting Test Data	260
Abbreviations in the Proof	261
Proving the Simple Matching Logic	262
Guidelines for Selecting Test Data	263
Discussion, Problems and Solutions	265
Topic 2 Complex Sequential Matching Logic	269
Orientation—Terminology and Objectives	269
The Reason for Complex Matching Logic	271
Pseudocode for Complex Matching Logic	271
Complex Matching Logic	273
Discussion, Problems and Solutions	280
Topic 3 Optional Matching Logic	284
Orientation—Terminology and Objectives	284
The Balance Line Method	286
A Less Desirable Method—for Comparison	290
Discussion, Problems and Solutions	296
Chapter 8 Indexed Files	297
Topic 1 COBOL Statements for Indexed Files	298
Orientation—Terminology and Objectives	298
Sequential Processing	298
COBOL Statements by Division	298
File Creation	302
A Sequential Update Program	306
Random Processing	318
COBOL Statements by Division	318
A Random Update Program	320
Discussion, Problems and Solutions	327
Topic 2 Multi-Indexed Files—Creation and Retrieval	337
Orientation—Terminology and Objectives	337
COBOL Statements for Multi-Indexed Files	337
Multi-Index Methods	341
Creating a Multi-Indexed File	341