COBOL
From Micro to Mainframe

Preparing for the New Millennium

ROBERT T. GRAUER
CAROL VAZQUEZ VILLAR
ARTHUR R. BUSS

THIRD EDITION

Micro Focus® Coverage Included

COBOL

From Micro to Mainframe

Preparing for the New Millennium

THIRD EDITION

ROBERT T. GRAUER

University of Miami

CAROL VAZQUEZ VILLAR

Andersen Consulting

ARTHUR R. BUSS

William Jewell College

PRENTICE HALL
Upper Saddle River, NJ 07458

Library of Congress Cataloging-in-Publication Data

Grauer, Robert T.
COBOL: from micro to mainframe; preparing for the new millennium, /Robert T. Grauer, Carol Vazquez Villar, Arthur R. Buss.
p. cm.
Includes index.
ISBN: 0-13-790817-2
1. COBOL (Computer program language). I. Vazquez Villar, Carol
II. Title.
CIP Data available.

To Marion, Benji, and Jessica
To Mario, Mom & Dad, and Fefa
To Pat, Betsy, and Carolyn

Acquisitions editor: **LAURA STEELE**
Publisher: **ALAN APT**
Editor-in-chief: **MARCIA HORTON**
Production editor: **IRWIN ZUCKER**
Managing editor: **BAYANI MENDOZA DE LEON**
Director of production and manufacturing: **DAVID W. RICCARDI**
Copy editor: **MARJORIE SHUSTAK**
Art director: **HEATHER SCOTT**
Cover designer: **JOHN CHRISTIANA**
Cover illustration: **PATRICE VAN ACKER**
Manufacturing buyer: **DONNA SULLIVAN**
Editorial assistant: **KATE KAIBNI**

Printed in the United States of America

10 9 8 7 6 5

ISBN 0-13-790817-2

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall of 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 Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

Preface

COBOL: From Micro to Mainframe 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 files and nonsequential files. The text also shows the new directions for COBOL in a chapter about Object-Oriented COBOL and an appendix devoted to the proposed changes in the COBOL 2000 standard.

All programs in the book can be run on personal computers or with minor modifications on mainframes or other platforms. The beauty of COBOL is that it can operate on any platform. This text provides instruction in ANS XOPEN standard COBOL. The one exception is the use of Micro Focus Object-Oriented COBOL in Chapter 20, since the final ANS standard has not yet been adopted.

Improvements in the Third Edition

The third edition responds to the requests of many students and instructors to provide access to Windows-based tools while maintaining the proven approach to teaching COBOL. Features of this edition include the following:

- The text has been modified to show the development of programs in a Windows environment. While the essential characteristics of COBOL remain unchanged, the development tools have not. This edition provides examples using one of the most popular Windows development tools available: Micro Focus® Personal COBOL™ for Windows™.
- Chapter 19 has been added to explain the Year 2000 problem. This chapter discusses the sources of the problem, shows why it is a problem, and discusses several techniques to correct the problem. At the end of the chapter, we provide a list of World Wide Web sites where further information may be obtained.
- Another new chapter (Chapter 20) demonstrates the concepts of Object-Oriented COBOL. This new approach to COBOL promises to be a way for companies to maintain the value of their legacy COBOL programs while still being able to use the benefits of object-oriented programming.
- Appendix A provides extensive coverage of the Micro Focus Personal COBOL for Windows. In addition to explaining every menu item and button, this appendix also includes a brief tutorial allowing the student to experience process of creating a program.
- Appendix B provides a guide to installing Personal COBOL for Windows and several techniques to make using the product easier.
- Coverage of COBOL 2000 and intrinsic functions has been added in Appendix E. The 1989 extensions to COBOL 85 allow the use of predefined functions that had been missing in COBOL. This appendix also discusses the changes anticipated in COBOL 2000.
- In Appendix G, there are 32 new projects *for student programming* assignments. Many of these projects build on previous tasks allowing the

student to experience the development of systems and the performance of maintenance.

- Various chapters have been changed to incorporate the changes in debugging and editing techniques used with a Windows programming environment as opposed to using a DOS compiler and debugger.

Benefits and Features

All of the features that have made the second edition successful have been retained and carried over into the third edition. These include:

- Immediate entry into COBOL programming, beginning in Chapter 1. Programming is learned by doing, and the book has students writing a complete program from the very beginning. Chapter 2 continues the discussion by having them execute the program of Chapter 1 in a thorough introduction to the programming process.
- Over 30 illustrative COBOL programs reinforce the discussion in the text and serve as both pedagogical aids and subsequent reference material. Every program is presented in a uniform and detailed format, including program narrative, record layouts, report layouts, test data, and processing specifications.
- A thorough discussion structured methodology, hierarchy charts, pseudocode, and top-down testing is presented in Chapter 3 and followed throughout. Students learn the proper way to develop programs early on and follow the procedure throughout the text.
- Every COBOL program in the text as well as data files for the student projects are available for download from a special World Wide Web site: http://www.prenhall.com/grauer_cobol.
- The availability of the sample listings enables students to reproduce and/or modify any of the programs without the tedium of data entry and further enhances the learning experience.
- An abundance of short-answer (true-false and fill-in) questions, COBOL problems, and programming projects for every chapter, with answers to the odd-numbered questions provided in Appendix F.
- Programming tips, dispersed throughout the text, that go beyond the syntactical rules of COBOL, and suggest stylistic considerations to make programs easier to read and maintain.
- Extensive use of graphic aids, featuring a two-color presentation, with figures to further clarify the presentation. Where Micro Focus Personal COBOL is discussed, actual pictures of the screens assist the student in understanding the user interface.
- System concept presentation at the beginning of most chapters, as COBOL instruction has come to require additional material beyond the language itself. There are detailed discussions of control breaks, data validation, techniques for table lookups and initialization, storing, the balance line algorithm for file maintenance and the organization of indexed files.
- While focusing on the proven techniques of structured programming and the established syntax of COBOL 85, the text also introduces the concepts of object-orientation and previews the significant changes *in COBOL 2000.*

Software and Supplements

The following software and supplements are available from Prentice Hall:

- SOFTWARE—Micro Focus Personal COBOL for Windows 3.1 with object-orientation and Personal Dialog System. Compatible with Windows95 and WindowsNT, Personal COBOL provides all the tools to help you learn and use COBOL. The software includes an integrated editor, compiler and animator for creating, debugging and executing COBOL programs. Prentice Hall offers an affordable package of *COBOL: From Micro to Mainframe, Third Edition* with the Micro Focus Personal COBOL Compiler. Please order ISBN 0-13-975178-5.
- WEB SITE—Download every COBOL program in the text as well as data files for the nearly on hundred student projects from the *COBOL: From Micro to Mainframe* web site at: http://www.prenhall.com/grauer_cobol.
- Instructor's Resource Manual (ISBN# 0-13-081513-6)
- Prentice Hall Custom Test. Based on the powerful testing technology developed by Engineering Software Associates, Inc. (EAS), Prentice Hall Custom Test allows the educator to create and tailor the exam to their own needs. Please order ISBN# 0-13-081515-2

Acknowledgments

We are especially grateful to our editors at Prentice Hall, Laura Steele, Alan Apt, and Marcia Horton, without whom this project would not have been possible. We also wan t to thank the many other individuals who helped produce the third edition. Irwin Zucker, who supervised the production, Kate Kaibni, editorial assistant, who worked hard to provide us with timely chapter reviews, and Joel Berman, our marketing manager at Prentice Hall, who developed the innovative campaign to make this book a success.

We also want to acknowledge our reviewers, who through their comments and constructive criticism, made this a far better book:

Robert V. Binder, Robert Binder Systems Consulting, Inc.
Dinon Boyer, University of Akron
Georgia Brown, Northern Illinois University
Jan De Lassen, Brigham Young University
Ida M. Flynn, University of Pittsburgh
Frank T. Gergelyi, NJIT
Ken Goldsmith, University of Miami
Tom Gorecki, St. Charles Community College
Carol C. Grimm, Palm Beach Community College
Monica Holmes, Central Michigan University
Ann W. Houck, Pima Community College
David Lee
James W. Payne, Kellogg Community College
Nicholas Ross, University of Illinois at Chicago
Wendell I. Pope, Utah State University
Daniel H. Rindfleisch, Computer Specialist with Federal Government
Daniel R. Rota, Robert Morris College
Richard H. Saracusa, Northeastern University
Ron Teemley, DeVry Institute of Technology
Donat Valcourt, Northeastern University
Ron Williams, McLennon Community College
Jackie Zucker, University of Miami

A final word of thanks to you, our readers, for choosing this book. Please feel free to contact us with any comments or suggestions via email.

Robert Grauer
rgrauer@umiami.miami.edu

Carol Vazquez Villar

Arthur R. Buss
bussa@william.jewell.edu

Contents

Chapter 4: The Identification, Environment, and Data Divisions 73

Chapter 5: The Procedure Division 97

Chapter 6: Debugging 139

Chapter 7: Editing and Coding Standards 169

Chapter 8: Data Validation 195

Chapter 9 : More About the Procedure Division 229

Chapter 16: Subprograms 475

Chapter 17: Sequential File Maintenance 515

Chapter 18: Indexed Files 549

1 Introduction

CHAPTER OUTLINE

OBJECTIVES

After reading this chapter you will be able to:

- Define the terms: field, record, and file.
- Name two techniques used to express program logic.
- Identify the four divisions of a COBOL program.
- State the six COBOL language elements.
- State the rules for creating a programmer-supplied name; distinguish between examples of valid and invalid names.
- State the difference between numeric and nonnumeric literals; recognize valid and invalid examples of each.
- Follow the logic of a simple program as expressed in a flowchart or pseudocode.

OVERVIEW

This book is about computer programming. In particular, it is about COBOL, a widely used commercial programming language. Programming involves the translation of an algorithm (a precise means of solving a problem) 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 and memory capacity.

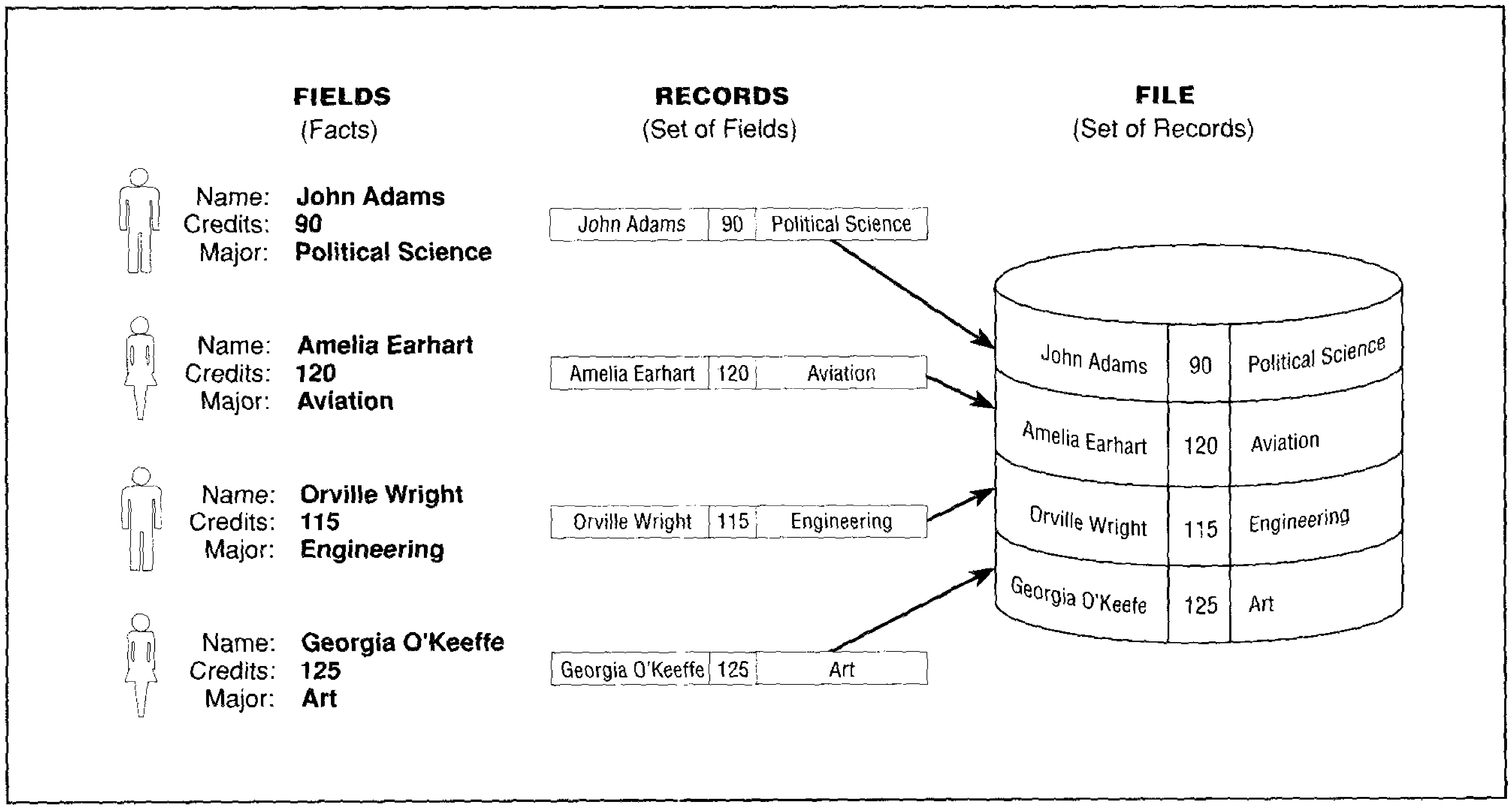
We begin our study of computer programming by describing a simple problem and then developing the logic and COBOL program to solve it. This rapid entrance into COBOL is somewhat different from the approach followed by most textbooks, but we believe in learning by doing. There is nothing very mysterious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

The First Problem

Our first problem is set in the context of a university, and involves a set of student records, one record per student. Each record contains the student's name, number of completed credits, and major. Implicit in this statement are the definitions of three fundamental terms: field, record, and file. A ***field*** is a basic fact, such as the name, address, major, grade point average, or number of completed credits. A ***record*** is a set of fields, and a ***file*** is a set of records. Thus, if there were 1,000 students, there would be 1,000 records (one for each student), each consisting of five fields, and comprising a single student file.

To clarify this relationship, we create four hypothetical students for our problem: John Adams, Amelia Earhart, Orville Wright, and Georgia O'Keeffe. There are many facts about each of our students, but our problem utilizes only three:

Figure 1.1 Fields, Records, and Files



name, major, and credits completed. Figure 1.1 represents these concepts in pictorial fashion. Each fact about each student comprises a single field. The three fields collectively make up that student's record. The four records (one for each of our students) compose the student file.

The problem is to process the file of student records and produce a list of engineering students who have completed more than 110 credits. It is a typical problem, in that its solution will address the three elements common to all computer applications: input, processing, and output. As shown in Figure 1.2, the student file, just defined, is the ***input***; this file is ***processed*** by determining which students are engineering majors with more than 110 credits; and consequently, a report is created as ***output***, reflecting these students.

The input to a computer program; that is, the precise arrangement of the various fields in each incoming record, has to be specified exactly. Figure 1.3a is a common way to communicate this information, and shows that the student's name is contained in positions 1–25, the number of credits in positions 26–28, and the student's major in positions 29–43. Note too, that every record in a given file must have the identical record layout.

In similar fashion, the report produced as output is also precisely designed. Figure 1.3c shows a print layout chart, in which descriptive information appears on line one, with the names of selected students in columns 9–33 of subsequent lines. Observe also that the location of the name field is different in the input and output records (positions 1–25 and 9–33, respectively), and that each input record contains three fields, but that each line of output has been designed to contain only one field.

Programming Specifications

It is important that programming specification—that is, *the input, processing, and output requirements*—be presented in a clear and unambiguous fashion.