Mastering

COBOL Programming

Palgrave Master Series

Accounting
Accounting Skills
Advanced English Language
Advanced English Literature
Advanced Pure Mathematics
Arabic
Basic Management
Biology
British Politics
Business Communication
Business Environment
C Programming
C++ Programming
Chemistry
COBOL Programming
Communication
Computing
Counselling Skills
Counselling Theory
Customer Relations
Database Design
Delphi Programming
Desktop Publishing
Economic and Social History
Economics
Electrical Engineering
Electronic and Electrical Calculations
Electronics
Employee Development
English Grammar
English Language
English Literature
Fashion Buying and Merchandising
Management
Fashion Styling
French

Geography
German
Global Information Systems
Human Resource Management
Information Technology
Internet
Italian
Java
Management Skills
Marketing Management
Mathematics
Microsoft Office
Microsoft Windows, Novell
NetWare and UNIX
Modern British History
Modern European History
Modern United States History
Modern World History
Networks
Organisational Behaviour
Pascal and Delphi Programming
Philosophy
Physics
Practical Criticism
Psychology
Shakespeare
Social Welfare
Sociology
Spanish
Statistics
Strategic Management
Systems Analysis and Design
Team Leadership
Theology
Twentieth Century Russian History
Visual Basic
World Religions

www.palgravemasterseries.com

Palgrave Master Series

Series Standing Order ISBN 978-0-333-69343-4

(outside North America only)

You can receive future titles in this series as they are published by placing a standing order. Please contact your bookseller or, in case of difficulty, write to us at the address below with your name and address, the title of the series and the ISBN quoted above.

Customer Services Department, Macmillan Distribution Ltd
Houndmills, Basingstoke, Hampshire RG21 6XS, England

Mastering

COBOL Programming

Second edition

Roger Hatty

*Head of Computer Science Department
School of Computing Sciences
De Montfort University
Leicester*

and

Mary Spence

*Senior Lecturer
Department of Information Systems
School of Computing Sciences
De Montfort University
Leicester*

Series Editor

Timothy Davies

Computer Consultant

palgrave

© Roger Hutter 1990, Roger Hutter and Mary Spence 1997

All rights reserved. No reproduction, copy or transmission of this publication may be made without written permission.

No paragraph of this publication may be reproduced, copied or transmitted save with written permission or in accordance with the provisions of the Copyright, Designs and Patents Act 1988, or under the terms of any licence permitting limited copying issued by the Copyright Licensing Agency, 90 Tottenham Court Road, London W1T 4LP.

Any person who does any unauthorised act in relation to this publication may be liable to criminal prosecution and civil claims for damages.

The authors have asserted their rights to be identified as the authors of this work in accordance with the Copyright, Designs and Patents Act 1988.

First edition 1983

Reprinted five times

Second edition 1997

Published by

PALGRAVE

Houndmills, Basingstoke, Hampshire RG21 6XS and

175 Fifth Avenue, New York, N. Y. 10010

Companies and representatives throughout the world

PALGRAVE is the new global academic imprint of
St. Martin's Press LLC Scholarly and Reference Division and
Palgrave Publishers Ltd (formerly Macmillan Press Ltd).

ISBN 978-0-333-68106-0 ISBN 978-1-349-14327-6 (eBook)

DOI 10.1057/9781349143276

This book is printed on paper suitable for recycling and made from fully managed and sustained forest sources.

A catalogue record for this book is available
from the British Library.

11	10	9	8	7	6	5	4	3
10	09	08	07	06	05	04	03	02

Contents

<i>Preface to the Second Edition</i>	<i>xiii</i>
<i>Acknowledgements</i>	<i>xv</i>
<i>List of Figures</i>	<i>xvi</i>

Part 1 NUCLEUS and SOURCE TEXT MANIPULATION modules

1 COBOL programming fundamentals	1
1.1 COBOL computer systems	1
1.2 COBOL 85 programs	5
1.2.1 A simple COBOL program	5
1.2.2 General structure of a COBOL program	5
1.2.3 General structure of a line of COBOL code	7
1.2.4 Use of upper or lower case for code	8
1.2.5 Use of the full stop (or period)	8
1.3 Program development	9
1.4 The IDENTIFICATION DIVISION	13
1.5 The ENVIRONMENT DIVISION	14
1.6 Chapter summary	15
1.7 Practical	16
2 DATA and PROCEDURE DIVISION fundamentals	17
2.1 The DATA DIVISION	17
2.2 Data item descriptions	18
2.2.1 Data names	18
2.2.2 The PICTURE clause	19
2.2.3 Data item level numbers	20
2.2.4 Data name qualification	22
2.2.5 The VALUE clause	23
2.2.6 Level 88 condition names	24
2.2.7 The USAGE clause	25
2.2.8 FILLER data names	26
2.2.9 Reference modification	26
2.2.10 The REDEFINES clause	27
2.3 The PROCEDURE DIVISION	28
2.4 Literals and figurative constants	30
2.5 The DISPLAY statement	31

2.6	COBOL syntax format	33
2.7	The MOVE statement	33
2.7.1	To set a data item to a value	33
2.7.2	To copy one data item to another	35
2.7.3	To edit and unedit data	36
2.7.4	To copy many items at a time	36
2.7.5	To change between USAGES	37
2.8	The INITIALIZE statement	37
2.9	Chapter summary	38
2.10	Practical	39
3	Numbers and the SUBTRACT statement	42
3.1	ADD statement variations	42
3.1.1	ADDing several numbers	42
3.1.2	The GIVING option	44
3.2	Fractional numbers	46
3.2.1	The V PICTURE	46
3.2.2	The P PICTURE	47
3.2.3	The . PICTURE	48
3.3	Signed numbers	50
3.3.1	The S PICTURE	50
3.3.2	The + and – PICTURES	52
3.3.3	The CR and DB PICTURES	53
3.4	The SUBTRACT statement	54
3.5	The ROUNDED option	56
3.6	The SIZE ERROR option	57
3.7	Chapter summary	61
3.8	Practical	62
4	The MULTIPLY, DIVIDE and COMPUTE statements, further data editing	64
4.1	The MULTIPLY statement	64
4.2	The DIVIDE statement	65
4.2.1	The basic DIVIDE statement	66
4.2.2	The REMAINDER option	66
4.3	The arithmetic statements together	67
4.4	The COMPUTE statement	70
4.4.1	The basic COMPUTE statement and arithmetic operators	70
4.4.2	Order of precedence of the arithmetic operators	71
4.4.3	The ROUNDED and SIZE ERROR options	72
4.4.4	Example program, using exponentiation	72
4.5	Zero suppression using Z and *	74
4.6	Insertion pictures , B 0 / and \$	75
4.7	Floating pictures \$ + -	76
4.8	The currency symbol and the decimal point	77
4.9	Chapter summary	78
4.10	Practical	79
5	The IF, GO TO and EVALUATE statements	81
5.1	The IF statement	81
5.2	Simple conditions	84
5.2.1	Relation conditions	84

5.2.2	Sign conditions	85
5.2.3	Class conditions	86
5.2.4	Condition name conditions	87
5.3	Nested IF statements	88
5.4	The NEXT SENTENCE and CONTINUE phrases	91
5.5	Complex conditions	92
5.5.1	Conditions with OR	92
5.5.2	Conditions with AND	93
5.5.3	Conditions with NOT	94
5.5.4	Abbreviated conditions	94
5.5.5	Conditions with both AND and OR	95
5.5.6	Conditions with level 88s	95
5.5.7	General rules when mixing AND, OR, NOT	96
5.6	The GO TO statement	96
5.7	The EVALUATE statement	97
5.7.1	Simple EVALUATE	98
5.7.2	Extended EVALUATE	99
5.7.3	Use of TRUE, FALSE and level 88s	102
5.8	Statements and sentences	104
5.9	Collating sequences	105
5.10	Date and time	106
5.10.1	ACCEPTing the date and time	107
5.10.2	Determining the century	109
5.10.3	Date comparisons, age calculations, etc	109
5.10.4	Leap year checks	110
5.10.5	Converting day of week and month to words	111
5.10.6	Converting to a 12-hour clock	111
5.11	Chapter summary	111
5.12	Practical	112
6	PERFORMing	114
6.1	Modular programming	114
6.1.1	Using the PERFORM statement to organise a program	114
6.1.2	Using the PERFORM statement to reduce duplication of code	116
6.2	PERFORM ... TIMES	117
6.2.1	The basic statement	117
6.2.2	The statement with in-line coding	119
6.2.3	The statement with out-of-line coding	120
6.3	PERFORM ... UNTIL	121
6.3.1	The statement with in-line and out-of-line coding	121
6.3.2	The WITH TEST BEFORE/AFTER options	123
6.4	PERFORM ... VARYING	124
6.4.1	The basic statement	124
6.4.2	The WITH TEST BEFORE/AFTER options	126
6.4.3	Using the loop data item in the loop	126
6.4.4	Using a condition that does not reference the loop data item	128
6.5	The PERFORM stack	129
6.6	PERFORM ... THRU and EXIT	131
6.7	Data validation	133
6.7.1	Validation of data type	133
6.7.2	Validation of ranges or specific values	135

6.7.3	Complex validations	137
6.8	The PERFORM procedure structure	138
6.9	Debugging	140
6.10	Chapter summary	141
6.11	Practical	142
7	Lists	144
7.1	List definition	144
7.2	List input and output	146
7.3	List calculations	148
7.3.1	Calculating the average of a list of numbers	148
7.3.2	Calculating the standard deviation	149
7.3.3	Finding the highest and lowest entries	151
7.4	Sorting	152
7.5	Lists of lists	154
7.6	INDEXED BY	156
7.7	The AFTER phrase	158
7.8	Lists with VALUEs	158
7.9	Chapter summary	160
7.10	Practical	161
8	Tables	164
8.1	Table definition	164
8.2	Table searching	166
8.2.1	Searching for a unique entry	166
8.2.2	Searching for non-unique entries	168
8.3	Sorting tables	169
8.4	Look-up tables	171
8.5	The SEARCH statement	174
8.5.1	Searching for a unique entry	174
8.5.2	Searching for non-unique entries	175
8.5.3	SEARCH ALL and ordered tables	177
8.6	A table example	178
8.7	Initialising tables	183
8.8	Chapter summary	184
8.9	Practical	184
9	String manipulation	186
9.1	The STRING statement	186
9.1.1	Eliminating spare characters	187
9.1.2	The OVERFLOW phrase	189
9.1.3	The POINTER phrase	190
9.2	The UNSTRING statement	191
9.2.1	Dividing a simple string	192
9.2.2	The DELIMITED BY phrase	192
9.2.3	The DELIMITER, COUNT, POINTER, TALLYING and OVERFLOW phrases	193
9.3	The INSPECT statement	196
9.3.1	The TALLYING phrase	196
9.3.2	The REPLACING phrase	199
9.3.3	Using TALLYING and REPLACING together	200
9.3.4	The CONVERTING phrase	200

9.4	Chapter summary	202
9.5	Practical	203
10	Source text manipulation	205
10.1	The COPY statement	205
10.1.1	Simple COPY statement	205
10.1.2	Nested COPY statement	207
10.1.3	Using a library	207
10.2	COPY and the REPLACING option	208
10.3	The REPLACE statement	209
10.4	Chapter summary	210
10.5	Practical	211
 Part II The INPUT-OUTPUT modules		
11	File concepts	215
11.1	File organisation	215
11.2	File access	217
11.3	File status	217
11.4	File processing	217
11.5	File definition	218
11.6	Record definition	220
11.7	The OPEN and CLOSE statements	221
11.8	The WRITE statement	222
11.9	The READ statement	225
11.10	File error handling	226
11.10.1	The USE statement	227
11.10.2	Using the file status value	228
11.11	File back-ups	229
11.12	Chapter summary	232
11.13	Practical	233
12	Processing sequential files	234
12.1	Searching	235
12.2	Amending a record	236
12.3	Record insertion	239
12.4	Record deletion	242
12.4.1	Using a delete flag	242
12.4.2	Using two files	243
12.5	Full sequential update	243
12.6	Dynamic allocation of file names	243
12.7	Printing a file	244
12.7.1	A simple file-listing	244
12.7.2	The ADVANCING clause	247
12.7.3	The LINAGE clause	247
12.7.4	A more sophisticated listing	248
12.8	Chapter summary	252
12.9	Practical	253
13	Processing relative files	254
13.1	Relative file attributes	254
13.2	OPENing and CLOSEing	256

13.3	Adding a record	257
13.4	Amending a record	259
13.5	Deleting a record	261
13.6	Retrieving records	262
13.6.1	Retrieving by the relative key	262
13.6.2	Retrieving by another record field	262
13.7	Sequential processing of the file	263
13.8	Using the START statement	263
13.9	The INVALID KEY clause	264
13.10	Chapter summary	265
13.11	Practical	266
14	Processing indexed files	268
14.1	Indexed file attributes	269
14.2	OPENing and CLOSEing	271
14.3	Adding a record	272
14.4	Amending a record	274
14.5	Deleting a record	276
14.6	Retrieving records	277
14.6.1	Retrieving by the prime record key	278
14.6.2	Retrieving by a secondary key	278
14.6.3	Retrieving by another record field	279
14.7	Sequential processing of the file	279
14.8	Using the START statement	281
14.8.1	STARTing at the beginning of a file	281
14.8.2	STARTing in the middle of a file	282
14.8.3	STARTing at the end of a file	282
14.9	The INVALID KEY clause	282
14.10	Changing indexed file attributes	283
14.11	Chapter summary	283
14.12	Practical	283
15	SORTing and MERGEing	285
15.1	The SORT statement	285
15.2	A simple SORT example	287
15.3	SORT INPUT and OUTPUT PROCEDURES	289
15.4	SORTing a list or table	292
15.5	MERGEing	294
15.6	Chapter summary	295
15.7	Practical	297

Part III The INTER-PROGRAM COMMUNICATION module

16	Communicating between programs	301
16.1	Basic subprogram features	301
16.2	Separately compiled subprograms	302
16.3	Parameter passing	304
16.3.1	Input parameters	304
16.3.2	Output parameters	305
16.3.3	An example with input and output parameters	306
16.4	Nested programs	308

16.4.1	General description of a nested program	308
16.4.2	Rules for calling subprograms	308
16.4.3	GLOBAL and local variables	310
16.4.4	An example nested program	310
16.5	Other subprogram features	315
16.5.1	The INITIAL clause	315
16.5.2	The CANCEL statement	315
16.5.3	The EXTERNAL clause	315
16.6	Testing with subprograms	316
16.6.1	The top-down approach	316
16.6.2	The bottom-up approach	317
16.7	Chapter summary	318
16.8	Practical	319

Part IV The REPORT WRITER module

17	Using the REPORT WRITER	323
17.1	The report	323
17.2	The ENVIRONMENT DIVISION entry	326
17.3	The DATA DIVISION entry	326
17.3.1	The RD clause	328
17.3.2	Specifying report lines	328
17.3.3	Page headings	328
17.3.4	Detail lines	329
17.3.5	Control headings	329
17.3.6	Control footings	329
17.3.7	Other facilities available	330
17.4	The PROCEDURE DIVISION entry	330
17.4.1	The INITIATE statement	331
17.4.2	The GENERATE statement	331
17.4.3	The TERMINATE statement	331
17.4.4	Empty reports	331
17.4.5	The SUPPRESS statement	332
17.4.6	USE BEFORE REPORTING	332
17.5	Another example program	332
17.6	Chapter summary	335
17.7	Practical	337

Part V The INTRINSIC FUNCTION module

18	Intrinsic FUNCTIONS	341
18.1	How the FUNCTIONS work	341
18.1.1	General format	344
18.1.2	Using alphabetic and alphanumeric functions	345
18.1.3	Using numeric and integer functions	345
18.1.4	Using multiple arguments, ALL table entries	345
18.1.5	Using a reference modifier	346
18.1.6	Nesting functions	346
18.2	Statistical FUNCTIONS	346
18.3	Date FUNCTIONS and REM	348
18.4	FUNCTION RANDOM	351

18.5	Chapter summary	352
18.6	Practical	352
Part VI Program debugging		
19	Program errors, testing and debugging	357
19.1	Some common errors	357
19.1.1	Using the arithmetic verbs	358
19.1.2	Conditions for branching and loops	359
19.1.3	Using EVALUATE	360
19.1.4	Data editing	361
19.1.5	Using tables	362
19.1.6	Sequential processing of a file	362
19.1.7	Writing reports	363
19.1.8	Direct processing of a file	365
19.2	Some testing strategies	366
19.3	Some items for a testing checklist	368
19.4	Chapter summary	369
Part VII Appendices		
Appendix A: COBOL 85 language formats		373
Appendix B: COBOL 85 reserved words		408
Appendix C: File status codes and meanings		411
Appendix D: Skeleton program		413
Appendix E: Standards for best practice		414
E.1	Program construction	414
E.2	Program code	416
E.3	Programmer-defined names	418
E.4	Some general points	419
Glossary of terms		421
Index		423

Preface to the Second Edition

COBOL, the COmmon Business Oriented Language, is one of many computer programming languages. It is the oldest Third Generation programming language still in use, but is still one of the most widely used. COBOL's popularity reflects the fact that a majority of computing applications are of a commercial nature and COBOL was specifically designed for use in commercial and business applications.

Compared with most other programming languages, COBOL has many facilities, some of which are quite sophisticated and of a relatively high level. This book covers almost all of the facilities specified in the ANSI COBOL 1985 standard, including the Nucleus, Sequential I-O, Relative I-O, Indexed I-O, Inter-Program Communication, Sort-Merge, Source Text Manipulation, Report Writer and Intrinsic Function modules, all at the highest level (Level 2). All the facilities are covered comprehensively.

This book is not only concerned with the rules of COBOL. There is also an art to programming: it is not enough to know the rules of COBOL – it is equally important to develop intelligible programs which have style and also use the best techniques available in the language. A professional and practical approach to programming is encouraged at all times throughout the book.

As with a foreign language, there is only one sure way to learn a programming language, and that is by exposure to many examples and plenty of practical experience. For this reason, this book contains over 70 examples of complete COBOL programs. Also with each program is a listing of a typical input and output of sample executions of the program. Additionally, at the end of each chapter, there are several directed practical experiences. Some of this practical work is provided by way of *trainers* (which are intended to help the reader to become familiar with the topics included in each chapter) and specified *programs* to write and test. Each chapter ends with a chapter summary.

The COBOL language material in this book has been organised into five *parts* – each part corresponding to a module (or group of modules) in the ANSI COBOL 85 standard. As far as possible, a deliberate attempt has been made to keep each part as independent as possible. However, of course, some parts rely upon some concepts covered in previous parts, but this has been kept to a minimum. This does mean that the parts do not necessarily have to be covered in the order in which they appear in the book. The book ends with a sixth part which discusses program testing and debugging.

As a further aid to learning, the material in the early part of the book has been organised so that COBOL programs can be written right from the start, from the first chapter onwards.

Although COBOL was originally designed as a programming language for *batch* computer systems (punched-card input and line-printer output), the language is being used more and more through a workstation (keyboard input and display output) such as a terminal connected to a computer or a stand-alone microcomputer. This book takes account of COBOL's use in the more recent workstation environment by assuming for most example programs that input is from a keyboard and output is to a display.

This book is suitable for COBOL courses in training centres and educational establishments (schools, colleges, polytechnics and universities) and for individuals who wish to learn the language by self-instruction. Also, practising COBOL programmers will find that this book is a useful reference source of the ANSI 1985 Standard for COBOL.

This second edition in the Mastering Series is a complete revision, by Mary Spence, of Roger Hutty's 'COBOL 85 Programming', and replaces the first edition. All chapters have been revised to make full use of COBOL 85 features, structured programming and the new Intrinsic Function Module that was implemented in 1989. A new Appendix has been added with suggestions for good practice for COBOL code.

The example program source code files are available using one of the following methods:

(a) File Transfer Protocol:

host name: ftp.dmu.ac.uk
log in as: anonymous
password: your email address
file: pub/cbl_comp/CBL-96.ZIP

(remember to set the file type to *binary*)
(unzip the files afterwards)

(b) Direct from Mary Spence at:

School of Computing Sciences
De Montfort University
The Gateway
Leicester LE1 9BH

enclosing a clean, unformatted disk and a stamped-addressed envelope.

Roger Hutty
Mary Spence

Acknowledgements

COBOL is an industry language and is not the property of any company or group of companies, or of any organisation or group of organisations.

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 authorised the use of this material in whole or in part, in the COBOL specifications. Such authorisation extends to the reproduction and use of COBOL specifications in programming manuals or similar publications.

(From the ANSI COBOL STANDARD X3.23 – 1985)

List of Figures

- 1.1 A basic COBOL computer system
- 1.2 A simple COBOL program
- 1.3 Structure diagrams for the AGE-CHECK program
- 1.4 The compilation and execution stages

- 2.1 A program to add 10 to a number
- 2.2 A program to add two numbers, using MOVE
- 2.3 A practical program to add numbers
- 2.4 A program to demonstrate USAGE

- 3.1 A program to add four numbers
- 3.2 A program to add three numbers using GIVING
- 3.3 The COST program
- 3.4 The BALANCE program
- 3.5 The DISCOUNT program
- 3.6 ADD with SIZE ERROR program
- 3.7 Structure diagram for ADD with SIZE ERROR program
- 3.8 Table for numeric editing
- 3.9 The SUBTRACT-PRACTICAL program

- 4.1 The INVESTMENT program
- 4.2 The COMPOUND-INTEREST program
- 4.3 The currency symbol specification
- 4.4 Editing character symbols
- 4.5 A program to experiment with editing

- 5.1 Structure diagram of the PAY-CALCULATION program
- 5.2 The PAY-CALCULATION program
- 5.3 The WORD-ORDER program
- 5.4 Structure diagram of the ELECTRICITY-BILL program
- 5.5 The ELECTRICITY-BILL program
- 5.6 Structure diagram for a case construct
- 5.7 The PROCESS-MARKS program
- 5.8 Decision table for the COLOUR program
- 5.9 The COLOUR program

- 5.10 A collating sequence specification
- 5.11 The TIME-AND-DATE program
- 5.12 Decision table logic to convert 24-hour clock time to 12-hour format

- 6.1 The PAY-MODULAR-VERSION program
- 6.2 The CHARACTER-DISPLAY program
- 6.3 The MORTGAGE program
- 6.4 The SHOPPING-BILL program
- 6.5 The NUMBERS-DISPLAY program
- 6.6 The TABLES program
- 6.7 The MORTGAGE-END program
- 6.8 The AGE-CHECK-2 program
- 6.9 The TABLES-2 program
- 6.10 The structure diagram of the TABLES-2 program

- 7.1 A list of five students' marks
- 7.2 The data structure diagram of mark-list
- 7.3 The MARKS-INPUT-OUTPUT program
- 7.4 The AVERAGE-MARKS program
- 7.5 The MARKS-STATISTICS program
- 7.6 The SORT-NAMES program
- 7.7 Students' exam paper marks
- 7.8 The PAPER-MARKS program
- 7.9 The DISPLAY-DAY program
- 7.10 Date validation logic structure diagram

- 8.1 A table of students' names and marks
- 8.2 The data structure diagram of student-table
- 8.3 The FIND-STUDENT'S-MARK program
- 8.4 The SORT-CLIENTS program
- 8.5 The DAYS-IN-MONTH program
- 8.6 The FIND-STUDENT'S-MARK-2 program
- 8.7 The DRINKS program
- 8.8 The WAGES program

- 9.1 The TV-RENTAL program
- 9.2 The STRING-WITH-POINTER program
- 9.3 The TV-RENTAL-INPUT program
- 9.4 The UNSTRING-INTO-LIST program
- 9.5 The UNSTRING-INTO-LIST-2 program
- 9.6 The SENTENCE-ANALYSIS program

- 10.1 Summary of COPY and REPLACE statements

- 11.1 Logical file organisations
- 11.2 File processing in a COBOL system
- 11.3 The WRITE-TO-FILE program
- 11.4 The READ-FROM-FILE program
- 11.5 The READ-FROM-FILE-2 program
- 11.6 The FILE-BACK-UP program
- 11.7 Basic file manipulation verbs

- 12.1 The STOCK-REORDER program
- 12.2 The STOCK PRICE CHANGE program
- 12.3 The STOCK-ITEM-INSERT program
- 12.4 The PRINT-A-FILE program
- 12.5 The PRINT-A-REPORT program
- 12.6 Permissible statements for sequential files

- 13.1 The PHONES-DIRECTORY programs (the first three DIVISIONS)
- 13.2 The PHONES-DIRECTORY program (main paragraphs)
- 13.3 The PHONES-DIRECTORY program (adding a record)
- 13.4 The PHONES-DIRECTORY program (amending a record)
- 13.5 The PHONES-DIRECTORY program (deleting a record)
- 13.6 Permissible statements for relative files (also applies to indexed files)

- 14.1 The PERSONNEL-FILE program (the first three DIVISIONS)
- 14.2 The PERSONNEL-FILE program (main paragraphs)
- 14.3 The PERSONNEL-FILE program (adding a record)
- 14.4 The PERSONNEL-FILE program (amending a record)
- 14.5 The PERSONNEL-FILE program (deleting a record)
- 14.6 The EMPLOYEES-BY-DEPARTMENT program

- 15.1 The SORT operation
- 15.2 A skeleton SORT program
- 15.3 The SORT-REGISTRATIONS program
- 15.4 The SORT-REGISTRATIONS-IO program
- 15.5 The SORT-CLIENTS-2 program
- 15.6 The MERGE-REGISTRATIONS program

- 16.1 The WELCOME-SCREEN subprogram
- 16.2 The TEST-WELCOME-SCREEN program
- 16.3 The WELCOME-SCREEN-2 subprogram
- 16.4 The TEST-WELCOME-SCREEN-2 program
- 16.5 Diagrammatic representation of a nested program
- 16.6 Diagrammatic representation of WAGES-PROGRAM-2
- 16.7 The WAGES-PROGRAM-2 program

- 17.1 The MONTHLY-WAGES-REPORT
- 17.2 The wages file
- 17.3 The MONTHLY-WAGES-REPORT program (the first two DIVISIONS)
- 17.4 The MONTHLY-WAGES-REPORT program (the DATA DIVISION)
- 17.5 The MONTHLY-WAGES-REPORT program (the PROCEDURE DIVISION)
- 17.6 The PRINT-A-REPORT-2 program

- 18.1 Table of FUNCTIONS
- 18.2 The MARKS-STATISTICS-2 program
- 18.3 The DATES program
- 18.4 The LUCKY-NUMBERS program

- 19.1 Do-ahead logic for sequential processing of a file
- 19.2 Do-ahead logic for a simple paged report