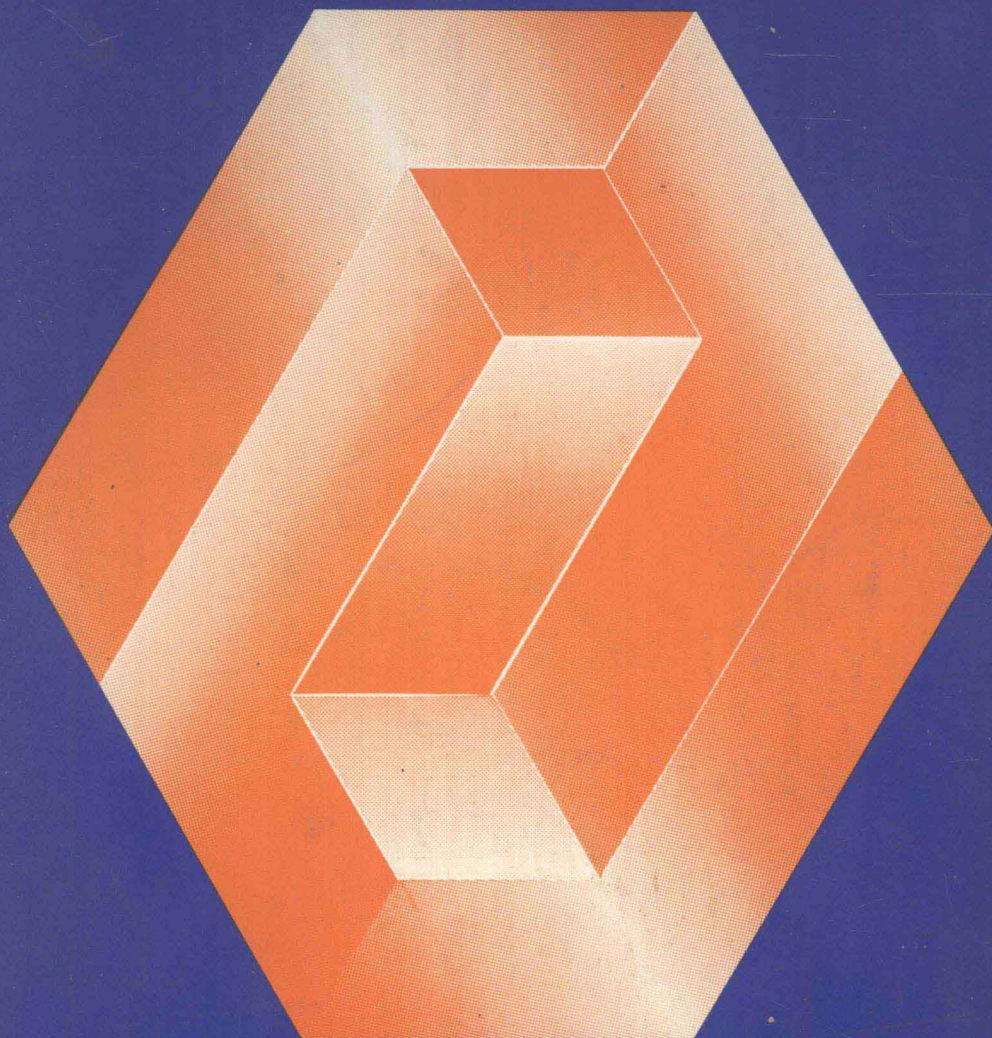


John B. Moore
Leo J. Makela

Structured FORTRAN with WATFIV



Text and Reference

Structured FORTRAN with WATFIV

Text and Reference

John B. Moore
Leo J. Makela

RESTON PUBLISHING COMPANY, INC., Reston, Virginia
A Prentice-Hall Company

Library of Congress Cataloging in Publication Data

Moore, John B ,
Structured Fortran with WATFIV.

Includes index.

1. FORTRAN (Computer program language) I. Makela,
Leo, joint author. II. Title.

QA76.73.F25M65 001.6'424 78-549

ISBN 0-8359-7101-5

©1978 by

RESTON PUBLISHING COMPANY, INC., Reston, Virginia

A Prentice-Hall Company

All rights reserved.

No part of this book may be reproduced in any way,
or by any means, without permission in writing from the publisher.

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America.

The Fortran programming language continues to be used extensively in North America as an aid in solving a wide variety of scientific and mathematical problems. Until recently however Fortran did not provide the programmer with many of the tools necessary to take advantage of the problem-solving approach which has come to be known as structured programming. Structured programming is primarily a method of organizing the task of algorithm development so that correct and efficient programs result. Its methods, along with those of hierarchical or top-down design, have helped transform the "art of computer programming" to the "science of computer programming". Several versions of Fortran now exist which contain the cornerstones of structured programming -- the IF-THEN-ELSE construct and the WHILE-DO construct. These, along with many other enhancements to standard Fortran, are now an integral part of the Fortran language processed by the WATFIV compiler developed at the University of Waterloo.

The purpose of this book is to provide a comprehensive description of the structured programming approach to problem-solving using the Fortran language processed by the WATFIV compiler. The authors believe it is the most complete and up-to-date book on the subject yet published.

The content is organized into four parts. Part I describes the statements in the language which define and manipulate fixed-point and floating-point values. In Part I the output device is always a printer, the input device, always a card reader. Unlike most programming texts which present new topics in the sequence: "Here's another statement in the language."; "Here's how it is used."; "Here's some examples.", the approach taken in this book is: "Here's what we are trying to do."; "Here's how we could do it using our existing knowledge."; "Here's an easier way of doing it.". In other words, the approach is to present the language in terms of purpose or function.

Part II describes the use of non-INTEGER, non-REAL values. Chapter 11 presents the concepts. Chapters 12 thru 16 each deal with a specific type of value. Chapter 17 concludes Part II with a summary of the rules for statements which may contain references to more than one type of value.

Part III describes the non-card input, non-printer output features of the language. Separate chapters describe

punched output, magnetic tape input-output, direct access input-output, and simulated input-output using CHARACTER variables. Two chapters describing NAMELIST input-output and unformatted READ-WRITE statements conclude Part III.

Part IV contains ten appendices. Aside from the standard topics usually found in programming texts, they include hints for program "debugging" and efficiency (Appendix G) and a comparison of the Fortran language processed by the WATFIV compiler and that processed by the IBM Fortran compilers (Appendix F). Flowcharting conventions and symbols are described in Appendix B. Flowcharts are used to define and illustrate the logic of the decision-making elements of the language. Appendix J describes how the structured language elements can be coded using the conventions of standard Fortran.

The text does not contain a description of how computers work. Many suitable books are available to the teacher who wishes to present these topics in parallel with a course on Fortran programming.

Use as a text. Because of the detailed nature of the explanations and example problems, the text can be used in a self-study environment. It is also appropriate for a one or two semester course. At Waterloo, first year students cover the material in Parts I and II in twenty-five lecture hours. They have an additional twenty-five hours of laboratory time during which tutorial help is available for assisting them with their assignments. Topics in Part I which are marked with an asterisk are omitted in this one semester course. The approach taken in lecture periods is to present one or two programming problems, introducing new language features as needed, but leaving the student to learn the details of the new statements on his own. More than five-hundred-and-fifty exercise, drill-type questions are included in the text. These provide the student with the opportunity for intensive practice on specific features of the language.

Use as a reference. Most introductory texts are unsuitable as a reference because the content is organized so that so-called "advanced" topics are presented toward the end of the book or as part of an appendix. This means that information on a particular topic may be found in two or more places thus necessitating much page-flipping and searching to find the required information. The fragmentation of topics is not present in this book. The index contains over 1000 references.

The text material was prepared, modified and printed using the SCRIPT text-editing system at the University of Waterloo. Texts, like programs, are not usually error-free. All programs used as examples have been carefully checked. Please accept our apologies for any undetected errors.

We acknowledge the contributions of both the Department of Applied Analysis and Computer Science and the Computer Systems Group and the University of Waterloo -- the developers of the WATFIV compiler. Specific thanks go to Sandra Ward and Bruce Hay for providing us with details of the most recent changes to the WATFIV language. Thanks also the many helpful suggestions from those who used the predecessor of this book, "WATFIV: Text and Reference", by John B. Moore.

To you the reader, We hope that you will find this a useful and enjoyable book. We have tried to keep your needs in mind from start to finish.

Waterloo, Ontario, Canada

John B. Moore
Leo J. Makela

To

Kelly, Jay, Greg, Mike,
the Koskinens and the Rivetts

This is a book that teaches you how to speak a language -- a language which permits you to use a computer to perform the numerical calculations necessary to solve problems. The language is one member of the Fortran family of languages. It was developed at the University of Waterloo to make using a computer easy for students. Some of you will learn to speak the language well -- others just enough to get by or to solve a specific problem. The more exercises and programming problems you do, the more fluent you will be in the use of the language. Good luck and good programming.

TABLE OF CONTENTS

PART I THE STATEMENTS IN THE LANGUAGE

CHAPTER 1 SO YOU WANT TO USE A COMPUTER

1.1	What Can A Computer Do?	3
1.2	Why Do We Use Them?	4
1.3	What Are The Five Steps In Problem Solving? ...	5
1.4	How About An Example?	8
1.5	What Happens When An Error Is Made?	20
1.6	A Second Example	24
1.7	A Third Example.	26
1.8	Summary	31

CHAPTER 2 NUMBERS, ARITHMETIC AND VARIABLES

2.1	INTEGER Values	33
2.2	REAL Numbers (Numbers with decimal points)	36
2.3	Operations Combining REAL And INTEGER Values ..	40
2.4	Variables	41
*2.5	Defining Your Own First-Letter Rule	44
2.6	Giving Values To Variables	46
2.7	Built-In Functions	53
2.8	Summary	59
2.9	Summary of WATFIV Statements Covered Thus Far .	60
2.10	Programming Problems	61

CHAPTER 3 DECISION AND CONTROL

3.1	Introduction	65
3.2	The IF-THEN-ELSE Construct	65
3.3	The Logical IF Statement	76
3.4	Executing One of Several Cases (DO CASE)	81
3.5	Non-Sequential Execution	83
*3.6	Other Explicit Control Statements	85
3.7	STOP, PAUSE and CONTINUE	91
3.8	Summary	92
3.9	Programming Problems	92

CHAPTER 4 LOOPS AND LOOP CONTROL

4.1	Loops and Loop Control	97
4.2	Program-Controlled Loops	97
4.3	WHILE-DO: A Summary	98
4.4	DO-Loops	100
4.5	Summary Of DO-Loop Rules	106
4.6	Nested Loops	109

4.7	Summary	113
4.8	Programming Problems	114

CHAPTER 5 INPUT

5.1	Input - The Three Questions	117
5.2	Format-Free Input	119
5.3	Formatted Input -The Concepts	124
5.4	Formatted Input of INTEGERS	125
5.5	Formatted Input of REAL Values (The F-Mask) ...	128
5.6	Formatted Input of REAL Values (The E-mask) ...	130
5.7	Input Containing REAL and INTEGER Values	131
5.8	A Second Method Of Positioning Masks	132
5.9	The X-Code	133
5.10	The Slash "/" Code	134
5.11	Two Handy Short Cuts	136
5.12	Two Unanswered Questions	138
5.13	Other Forms of the Formatted READ Statement ...	140
5.14	A Word About Commas	142
*5.15	Literals In Input FORMAT Statements	142
*5.16	The G-Mask	143
*5.17	F-Mask Modification - The P-Code	144
*5.18	The ERR=m Parameter	146
5.19	Summary	147
5.20	Programming Problems	148

CHAPTER 6 OUTPUT

6.1	Output - The Three Questions	151
6.2	Three Facts About Printed Output	153
6.3	Formatted Output - The Concepts	153
6.4	Formatted INTEGER Output	154
6.5	Vertical Control	158
6.6	A Second Method Of Horizontal Control	159
6.7	The X-Code	160
6.8	"Hiding" The VCC	161
6.9	Printing REAL Values	162
*6.10	The General Mask - The G-Mask	166
*6.11	Mask Modification - The P-Code	167
6.12	Literals	169
6.13	The Slash "/" Code	171
6.14	Two Short Cuts	172
6.15	Two Unanswered Questions	175
6.16	A Word About Commas	179
*6.17	Format-Free Output	180
6.18	Summary	181
6.19	Programming Problems	182

CHAPTER 7 SETS OF VARIABLES

7.1	It Looks Easy But ...	185
7.2	A Second Example Using Subscripted Variables ..	188
7.3	Valid and Invalid Uses	190
7.4	Input-Output of Subscripted Variables	192
7.5	Tables Of Variables	199
7.6	Higher Dimensioned Arrays	205
7.7	Assignment of Values at Compile Time	206
*7.8	The DATA Statement	208
*7.9	The DIMENSION Statement	210
7.10	Summary	211
7.11	Programming Problems	212

CHAPTER 8 REMOTE BLOCKS

8.1	A Simple Payroll Problem	217
8.2	Logic for a General READ loop	220
8.3	The WHILE-EXECUTE Statement	222
8.4	Rules for REMOTE BLOCK and END BLOCK	222
8.5	Summary	225
8.6	Programming Problems	225

CHAPTER 9 SUBPROGRAMS -- THE BASICS

9.1	A Problem Requiring Factorials	230
9.2	SUBROUTINE Subprograms	233
9.3	Basic Rules of SUBROUTINE Usage	240
9.4	FUNCTION Subprograms	243
9.5	Subprograms -- Rule Summary	248
9.6	Statement Functions	251
9.7	Summary	254
9.8	Programming Problems	255

CHAPTER 10 SUBPROGRAMS -- THE DETAILS

10.1	"Memory" Concepts	260
10.2	Value Assignments at CALL and RETURN Times ...	264
10.3	Execution-Time Dimensioning of Arrays	266
10.4	Communication Via A Common Block	273
10.5	Multiple Common Blocks	277
10.6	BLOCK DATA Subprograms	280
*10.7	EQUIVALENCE (Shared Storage Within A Segment) .	282
*10.8	Multiple Entries To Subprograms	288
*10.9	Multiple Returns From SUBROUTINES	291
*10.10	EXTERNAL Subprograms	292
10.11	Summary	295
10.12	Programming Problems	297

CHAPTER 11 OTHER TYPES OF VALUES - THE CONCEPTS

11.1	Other Types Of Variables	303
11.2	The Concept of "Length".	304
11.3	Summary	306

CHAPTER 12 INTEGER VARIABLES OF LENGTH TWO

12.1	Purpose	307
12.2	Defining INTEGER*2 Variables	307
12.3	Constants, Expressions, and Assignments	308
12.4	Built-In Functions	309
12.5	Input-Output Of INTEGER*2 Variables	309
12.6	Valid And Invalid Uses	309
12.7	Summary	310
12.8	Programming Problems	310

CHAPTER 13 EXTENDED PRECISION

13.1	Extended Precision Values	311
13.2	E.P. Constants, Variables and Expressions	311
13.3	E.P. Assignment Statements	315
13.4	Extended Precision Built-In Functions	317
13.5	Input-Output Of E.P. Values	319
13.6	Subprograms	320
13.7	Other Statements Involving E.P. Values	321
13.8	Summary	322
13.9	Programming Problems	322

CHAPTER 14 COMPLEX VALUES

14.1	Complex Numbers	325
14.2	Constants, Variables, and Expressions	325
14.3	COMPLEX Assignment Statements	328
14.4	COMPLEX Built-In Functions	328
14.5	Input-Output Of COMPLEX Values	330
14.6	Subprograms	332
14.7	Extended Precision COMPLEX Values	334
14.8	Other Statements Involving COMPLEX Values	337
14.9	Summary	337
14.10	Programming Problems	337

CHAPTER 15 LOGICAL VALUES

15.1	LOGICAL Values	341
15.2	LOGICAL Constants, Variables and Expressions	341
15.3	Assignment Statements and Comparisons	344

15.4	Built-In Functions	345
15.5	Input-Output of LOGICAL Values	345
15.6	Subprograms	346
15.7	LOGICAL*1 Variables	347
15.8	Summary	349
15.9	Programming Problems	349

CHAPTER 16 CHARACTER VALUES

16.1	CHARACTER Values	351
16.2	Constants, Variables, and Expressions	352
16.3	Comparison of CHARACTER Values	353
16.4	CHARACTER Assignment Statements	354
16.5	Built-In Functions	355
16.6	Input-Output of CHARACTER Values	355
16.7	Subprograms	358
16.8	Two Example Programs	359
16.9	Storing Format Codes In CHARACTER Arrays	360
16.10	Hexadecimal Values	365
16.11	Bit Assignment And Manipulation	368
16.12	Other Symbols In WATFIV	373
16.13	Other Statements Involving CHARACTER Values ..	374
16.14	Summary	374
16.15	Programming Problems	375

CHAPTER 17 MIXED-TYPE STATEMENTS -- A SUMMARY

17.1	Allowed Types And Length of Variables	377
17.2	Specification Of Variable Type	378
17.3	Defining Arrays	379
17.4	Specification Of Initial Values	380
17.5	Mixed-Type Expressions And Statements	380
17.6	Subscript Values	382
17.7	Shared Storage Within A Segment(EQUIVALENCE) .	383
17.8	Shared Storage Among Segments(COMMON)	385
17.9	Argument And Parameter Lists	386
17.10	FUNCTION Subprograms	386
17.11	Input-Output	387
17.12	Storing CHARACTER Values In Other Variables ..	388
17.13	Ordering Of Statements In Program Segments ...	390
17.14	Summary	391

CHAPTER 18 PUNCHED OUTPUT

18.1	Punched Output	395
18.2	Formatted Punched Output	396
18.3	Format-Free Punched Output	397
18.4	An Example Program	397
18.5	Summary	399
18.6	Programming Problems	399

CHAPTER 19 MAGNETIC TAPE INPUT-OUTPUT

19.1	Magnetic Tape And Magnetic Tape Units	401
19.2	A First Example	402
19.3	Unformatted Input-Output	404
19.4	Formatted Input-Output	409
19.5	Format-Free Input-Output	412
19.6	Other Topics	413
19.7	Summary	413
19.8	Programming Problems	416

CHAPTER 20 DIRECT ACCESS INPUT OUTPUT

20.1	Direct Access Concepts	419
20.2	Direct Access In WATFIV	419
20.3	Unformatted Direct Access Input-Output	420
20.4	Summary Of Unformatted Input-Output	422
20.5	Formatted Input-Output	423
20.6	Summary Of Formatted Input-Output	427
20.7	Input-Output Using A Mix Of Statements	429
20.8	The FIND Statement	429
20.9	Other Topics	430
20.10	Summary	432
20.11	Programming Problems	433

CHAPTER 21 CHARACTER "DEVICES"

21.1	Concepts And Uses Of CHARACTER Devices	435
21.2	CHARACTER Variables Used As Input Devices	436
21.3	Use Of CHARACTER Input Devices	438
21.4	CHARACTER Variables As Output "Devices"	443
21.5	Print-Line Images	446
21.6	Execution-Time Formatting	449
21.7	Summary	456
21.8	Programming Problems	457

CHAPTER 22 NAMELIST AND DUMPLIST

22.1	NAMELIST Output	459
22.2	NAMELIST Input	461
22.3	DUMPLIST Output	463
22.4	Summary	464
22.5	Programming Problems	464

CHAPTER 23 UNFORMATTED INPUT-OUTPUT

23.1	Unformatted Input-Output	465
23.2	Other Facts About Unformatted Input-Output ...	466
23.3	Summary	468
23.4	Programming Problems	468

PART IV

APPENDICES

APPENDIX A	THE PUNCHED CARD	471
APPENDIX B	FLOWCHARTING	477
APPENDIX C	INTERNAL REPRESENTATION OF VALUES	481
APPENDIX D	SYMBOLS FOUND ON PRINT TRAINS	487
APPENDIX E	JOB CONTROL STATEMENTS	489
APPENDIX F	WATFIV - FORTRAN COMPARISONS	497
APPENDIX G	DEBUGGING HINTS, EFFICIENCY, GENERALITY	501
APPENDIX H	WATFIV Messages	507
APPENDIX I	CONVERSION TO NON-STRUCTURED FORTRAN	521
APPENDIX J	BUILT-IN FUNCTIONS	525
INDEX		529

Part I

THE STATEMENTS IN THE LANGUAGE

