Fortran for Students

Macmillan Computer Science Series

Consulting Editor
Professor F. H. Sumner, University of Manchester

G. M. Birtwistle, *Discrete Event Modelling on Simula*

J. K. Buckle, *The ICL 2900 Series*

Derek Coleman, *A Structured Programming Approach to Data* *

Andrew J. T. Colin, *Programming and Problem-solving in Algol 68* *

S. M. Deen, *Fundamentals of Data Base Systems* *

David Hopkin and Barbara Moss, *Automata* *

Roger Hutty, *Fortran for Students*

H. Kopetz, *Software Reliability*

A. Learner and A. J. Powell, *An Introduction to Algol 68 through Problems* *

A. M. Lister, *Fundamentals of Operating Systems, second edition* *

Brian Meek, *Fortran, PL/I and the Algols*

Derrick Morris and Roland N. Ibbett, *The MU5 Computer System*

John Race, *Case Studies in Systems Analysis*

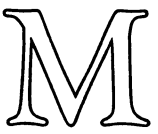
I. R. Wilson and A. M. Addyman, *A Practical Introduction to Pascal*

* The titles marked with an asterisk were prepared during the Consulting Editorship of Professor J. S. Rohl, University of Western Australia.

Fortran for Students

Roger Hutty

School of Mathematics, Computing and Statistics,
Leicester Polytechnic



© Roger Hutty 1980

All rights reserved. No part of this publication may be reproduced or transmitted, in any form or by any means, without permission.

First published 1980 by
THE MACMILLAN PRESS LTD
London and Basingstoke
Associated companies in Delhi Dublin
Hong Kong Johannesburg Lagos Melbourne
New York Singapore and Tokyo

ISBN 978-1-349-81447-3 ISBN 978-1-349-81445-9 (eBook)
DOI 10.1007/978-1-349-81445-9

This book is sold subject to the standard conditions of the Net Book Agreement.

The paperback edition of this book is sold subject to the condition that it shall not, by the way of trade or otherwise, be lent, resold, hired out, or otherwise circulated without the publisher's prior consent in any form of binding or cover other than that in which it is published and without a similar condition including this condition being imposed on the subsequent purchaser.

Contents

	Preface	vii
1	Introduction	1
	1.1 Computer Systems	1
	1.2 Programming	1
	1.3 A simple FORTRAN program	3
	1.4 Real constants and variables	5
	1.5 Arithmetic assignment statements	6
	Program	8
2	Integers, the IF and GOTO statements	9
	2.1 Integer constants and variables	9
	2.2 Integer arithmetic assignment statements	9
	2.3 Integer input and output	10
	2.4 IF and GOTO statements and relational expressions	11
	2.5 Text output	15
	2.6 Statements	15
	2.7 FORTRAN 77	16
	Program	18
3	Looping	19
	3.1 Looping and the DO and CONTINUE statements	19
	3.2 Output layout	24
	3.3 FORTRAN 77	25
	Program	26
4	Arrays	28
	4.1 Arrays	28
	4.2 Use of / in a FORMAT statement	31
	4.3 Implied-DO lists	33
	4.4 FORTRAN 77	34
	Program	35

5 Functions 36

5.1 Basic external functions 36
5.2 Intrinsic functions 37
5.3 Successive approximations 38
5.4 Nested DO loops 39
5.5 Array subscript expressions 40
5.6 Sorting 40
5.7 Decimal exponent values 42
5.8 FORTRAN 77 42
Program 43

6 Two-dimension arrays 44

6.1 Two-dimension arrays 44
6.2 Input and output of 2-D arrays 46
6.3 Group format specification 47
6.4 FORTRAN 77 48
Program 48

7 Procedures 50

7.1 Procedure concepts 50
7.2 Statement functions 50
7.3 Function subprograms 53
7.4 Subroutine subprograms 55
7.5 Comparison and use of procedures 57
Program 58

8 Character handling, DATA and Type-statements 60

8.1 Character handling 60
8.2 The DATA statement 62
8.3 Type-statements 64
8.4 Precision 65
8.5 FORTRAN 77 66
Program 68

9 The Computed GOTO, logical and Arithmetic IF statements 70

9.1 The Computed GOTO statement 70
9.2 Logical expressions 72
9.3 Logical statements 74
9.4 Input and output of logical variables 75
9.5 The Arithmetic IF statement 76
9.6 FORTRAN 77 78
Program 79

10 The COMMON and EQUIVALENCE statements 81

10.1 The EQUIVALENCE statement 81
10.2 The COMMON statement 82

Exercise Answers 85

Index 97

Preface

This learning text, based on FORTRAN 66, - the ANSI X3.9-1966 FORTRAN standard enables a student to develop through suitable exercises and programming practice a professional approach to programming and produce a work of quality. There is an art to programming: it is not enough for a student to know the rules of a language, it is equally important to develop a readable and intelligible program which has style and uses the best techniques.

The text is in the form of short narratives; each narrative covers one or two concepts and is followed by an exercise. The narratives move fast, the exercises are challenging and they encourage students to make many discoveries for themselves, thereby allowing the rapid development of skills. Every opportunity is taken to expose practical problems and develop good programming habits right from the start.

The text includes some opinions and opposing views for discussion.

Most of the facilities specified in the FORTRAN 66 standard are included in the text.

The main changes specified by the FORTRAN 77 (ANSI X3.9-1978) standard are included in the text as amendments at the end of each chapter. This way of including FORTRAN 77 was chosen because it will be several years before all of the facilities specified in the standard are available on all computer systems.

The text can be used with the computer systems of all the major manufacturers.

It is suitable for BSc and HND courses in Computing, Mathematics and Engineering and as a reference book for practising FORTRAN programmers.

NOTES FOR TUTORS

The text is divided into chapters. A chapter can be used as a self-contained unit for teaching purposes to be used for one or two weeks' lectures and tutorials or one self-instructional unit.

Each chapter assumes a student has satisfactorily completed the previous chapters.

The book is not an ordinary text book - it is a learning text. The narratives, the exercises, the exercise answers and the program at the end of each chapter all play an equally important part in the learning process. The narrative is fast moving. The exercises have been chosen to be a real test of the student's understanding of both concept and detail and to extend, sometimes by discovery, the knowledge and understanding gained in the narrative. Some of the exercise answers include notes on the answers which also extend a student's knowledge and understanding. A program is specified at the end of each chapter for the student to code, run and test on a computer system. Each program is designed, as far as is possible, to include the concepts contained in the chapter.

Students should be encouraged to do the exercises when they are encountered and before continuing with the text. Additionally, the programs should at least be coded before continuing with the next chapter.

Many students will be able to work through the text without assistance, allowing the tutor to concentrate his attention on those students who are unable to complete the text on their own.

The whole text is designed on the premise that one sure way to learn a programming language is by plenty of practical experience. That is one reason why practical exercises have been inserted throughout the text and a program included at the end of each chapter. The text is also designed so that students can code and run programs at the earliest possible stage in a course.

Only one program is specified at the end of each chapter so that the student can concentrate his effort on that one program. It is important that the student be encouraged to produce a fully correct working version of each program - one which adheres exactly to the program specification. Students should present only correctly working programs for assessment so that the tutor can concentrate on the style and technique of the programs. If necessary, the programs can be replaced by other programs specified by the tutor for specialist courses, although the programs in the text have been carefully chosen to cover as many aspects of computing as possible.

Instead of covering all of a topic at once, most topics are spread over a few chapters and each chapter covers parts of a few topics. This enables students to write more relevant programs earlier in a course and it makes the text more interesting thereby helping a student's concentration. Additionally, because a topic is spread throughout the text, a student has time gradually to assimilate the concepts contained within a topic and to absorb each concept before proceeding to the next.

The text is machine independent and follows the FORTRAN 66 standard. Where the standard allows for interpretation by computer manufacturers, common variations are treated. The COMMON and EQUIVALENCE statements are covered only briefly in the text because of their higher level nature and complexity in being used correctly. It is felt, also, that these two statements and the facilities which they provide are best explained in a 'chalk and talk' situation. The COMPLEX facility is not covered in the book because of its specialist nature.

The FORTRAN 77 standard amendments appear at the end of each chapter. Some computer systems already support some of the amendments. It will take several years for all computer systems to have full FORTRAN 77 standardisation. In the meantime, tutors should inform their students which facilities specified in the text are currently available on the computer system they are using.

One problem encountered by tutors teaching programming languages to a group of students is the wide variation in the students' rates of learning the language - even groups of students with the same background. Consequently, a favoured method of teaching programming languages is by tutor-assisted self-instructional methods, so that students may learn at their own rate and not become bored or lost. This text is particularly suitable for such a course - in fact, part of the text has been used as such for a few years.

It will be necessary for the tutor to introduce the students to the particular computer they will be using and to tell them the procedures for using it. If the computer system provides an equivalent choice between batch and terminal use, students should use the batch initially because, although terminals certainly motivate students, they also encourage very poor programming style and technique. Students can equally well be motivated by a good batch facility.

NOTES FOR STUDENTS

The text is not intended as a beginners' teach-yourself text. It is intended to accompany a course of lectures or be used as the text for a tutor-assisted self-instructional course.

However, it is suitable as a teach-yourself text for students who already have an appreciation of computer programming, or who already know a little FORTRAN. Practising FORTRAN programmers may find the text useful for reference purposes, especially the FORTRAN 77 amendments at the end of each chapter.

There is only one way to learn a programming language thoroughly and that is to have plenty of experience in using it - just like learning a foreign language. Proficiency in FORTRAN is acquired by writing it. This text contains many exercises which call for written answers; it is essential that you do write down the answers. Eager students are sometimes tempted to answer the questions mentally in their impatience to make progress with the subject. Resist this temptation!

All exercises should be attempted as you encounter them and before continuing with the text. The answers to exercises should be written down and checked with the answers given at the end of the book. Some answers include additional information.

If you are unable to answer a question, read the relevant parts of the text again. If you get stuck, or if there is a point that is not clear, ask your tutor.

At the end of each chapter there is a program specification. The program should be coded, run and tested, preferably before passing on to the next chapter, although this may not be practicable; however, the program should at least be coded before looking at the next chapter. The program should be a fully correct working version which adheres exactly to the specification. A program is no use to anyone if it does not do exactly what is required of it.

The main text adheres to the FORTRAN 66 standard. Amendments specified in the later FORTRAN 77 standard are included at the end of each chapter. The computer system you are using may not support all, or any of the amendments - you should check with your tutor which facilities are available on your computer system.

ACKNOWLEDGEMENTS

My thanks to Arthur Radford, for checking my FORTRAN; Roger Barnes, for checking my English; Peter Leadbetter for checking the final text; Leicester Polytechnic Computer Centre for allowing me to use their computer facilities for the testing and production of the programs in the book; and last, but not least, my wife, Susan, for her support in many ways, particularly in the typing of the draft and final copy of the text.

Roger Hutty