Macmillan Computer Science Series

Consulting Editor

Professor F. H. Sumner, University of Manchester

S. T. Allworth and R. N. Zobel, *Introduction to Real-time Software Design, second edition*

Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using FORTRAN 77*

Ian O. Angell and Gareth Griffith, *High-resolution Computer Graphics Using Pascal*

M. A. Azmoodeh, *Abstract Data Types and Algorithms*

Philip Barker, *Author Languages for CAL*

A. N. Barrett and A. L. Mackay, *Spatial Structure and the Microcomputer*

R. E. Berry and B. A. E. Meekings, *A Book on C*

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

T. B. Boffey, *Graph Theory in Operations Research*

Richard Bornat, *Understanding and Writing Compilers*

Linda E. M. Brackenbury, *Design of VLSI Systems – A Practical Introduction*

J. K. Buckle, *Software Configuration Management*

W. D. Burnham and A. R. Hall, *Prolog Programming and Applications*

J. C. Cluley, *Interfacing to Microprocessors*

Robert Cole, *Computer Communications, second edition*

Derek Coleman, *A Structured Programming Approach to Data*

Andrew J. T. Colin, *Fundamentals of Computer Science*

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

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

S. M. Deen, *Principles and Practice of Database Systems*

Tim Denvir, *Introduction to Discrete Mathematics for Software Engineering*

P. M. Dew and K. R. James, *Introduction to Numerical Computation in Pascal*

M. R. M. Dunsmuir and G. J. Davies, *Programming the UNIX System*

K. C. E. Gee, *Introduction to Local Area Computer Networks*

J. B. Gosling, *Design of Arithmetic Units for Digital Computers*

Roger Hutton, *Fortran for Students*

Roger Hutton, *Z80 Assembly Language Programming for Students*

Roland N. Ibbett, *The Architecture of High Performance Computers*

Patrick Jaulent, *The 68000 – Hardware and Software*

J. M. King and J. P. Pardoe, *Program Design Using JSP – A Practical Introduction*

H. Kopetz, *Software Reliability*

E. V. Krishnamurthy, *Introductory Theory of Computer Science*

V. P. Lane, *Security of Computer Based Information Systems*

Graham Lee, *From Hardware to Software – an introduction to computers*

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

G. P. McKeown and V. J. Rayward-Smith, *Mathematics for Computing*

(continued overleaf)

Brian Meek, *Fortran, PL/1 and the Algols*
 Barry Morrell and Peter Whittle, *CP/M 80 Programmer's Guide*
 Derrick Morris, *System Programming Based on the PDP11*
 Pim Oets, *MS-DOS and PC-DOS – A Practical Guide*
 Christian Queinnec, *LISP*
 E. J. Redfern, *Introduction to Pascal for Computational Mathematics*
 Gordon Reece, *Microcomputer Modelling by Finite Differences*
 W. P. Salman, O. Tisserand and B. Toulout, *FORTH*
 L. E. Scales, *Introduction to Non-linear Optimization*
 Peter S. Sell, *Expert Systems – A Practical Introduction*
 Colin J. Theaker and Graham R. Brookes, *A Practical Course on Operating Systems*
 J-M. Trio, *8086-8088 Architecture and Programming*
 M. J. Usher, *Information Theory for Information Technologists*
 B. S. Walker, *Understanding Microprocessors*
 Peter J. L. Wallis, *Portable Programming*
 Colin Walls, *Programming Dedicated Microprocessors*
 I. R. Wilson and A. M. Addyman, *A Practical Introduction to Pascal – with BS6192, second edition*

Non-series

Roy Anderson, *Management, Information Systems and Computers*
 J. E. Bingham and G. W. P. Davies, *A Handbook of Systems Analysis, second edition*
 J. E. Bingham and G. W. P. Davies, *Planning for Data Communications*

Introduction to Pascal for Computational Mathematics

E. J. Redfern

*Department of Statistics
Leeds University*

M
MACMILLAN
EDUCATION

© E. J. Redfern 1987

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 Act 1956 (as amended), or under the terms of any licence permitting limited copying issued by the Copyright Licensing Agency, 33-4 Alfred Place London WC1E 7DP.

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

First published 1987

Published by
MACMILLAN EDUCATION LTD
Houndmills, Basingstoke, Hampshire RG21 2XS
and London
Companies and representatives
throughout the world

Typeset by TecSet Ltd, Wallington, Surrey

British Library Cataloguing in Publication Data
Redfern, E. J.

Introduction to Pascal for computational
mathematics.—(Macmillan computer science
series).

1. Mathematics—Data processing 2. PASCAL
(Computer program language)

I. Title

510'.28'55133 QA76.95

ISBN 978-0-333-44431-3 ISBN 978-1-349-18977-9 (eBook)

DOI 10.1007/978-1-349-18977-9

Contents

<i>Preface</i>	ix
1 Introduction	1
2 First Steps	6
2.1 Introduction	6
2.2 A closer look	9
2.3 The assignment statement	15
2.4 Standard functions	17
2.5 Constants	19
2.6 Comments	20
2.7 Compiling and running a program	20
2.8 Exercises	22
3 Programming Structures	25
3.1 Designing a program – the algorithm	25
3.2 Control statements	28
3.3 Looping	34
3.4 Nesting of control statements	39
3.5 Exercises	41
4 Sequences and Series	43
4.1 Introduction	43
4.2 Sequences	44
4.3 Series	50
4.4 Exercises	55
5 Numerical Errors and Accuracy	58
5.1 Introduction	58
5.2 Representation of real numbers	58
5.3 Computer arithmetic	60
5.4 Order of calculation	65
5.5 Exercises	67

6	More on Entering Data and Displaying Results	70
6.1	Introduction	70
6.2	Output to files	71
6.3	Input from files	72
6.4	Construction of tables	75
6.5	Exercises	78
7	Procedures and Functions	79
7.1	Introduction	79
7.2	Procedures	79
7.3	Functions	83
7.4	Nesting of procedures and functions	85
7.5	Structured program design	86
7.6	Recursion	90
7.7	Exercises	93
8	Iterative Procedures	94
8.1	Iteration	94
8.2	Programming an iterative procedure	95
8.3	Newton-Raphson	98
8.4	The secant method	101
8.5	Functions and procedures as parameters	104
8.6	Exercises	105
9	Numerical Calculus	106
9.1	Numerical differentiation	106
9.2	Numerical integration	107
9.3	Simpson's rule	112
9.4	Errors in numerical integration	114
9.5	Exercises	116
10	First-order Differential Equations	118
10.1	Introduction	118
10.2	Euler's method	118
10.3	Modified Euler	119
10.4	Example	120
10.5	Programming aspects	122
10.6	Predictor-corrector methods	123
10.7	Comment	126
10.8	Exercises	126

11	Arrays	129
11.1	Vectors or one-dimensional arrays	129
11.2	Specification of subscript ranges	133
11.3	Multi-dimensional arrays	133
11.4	Passing arrays to procedures	134
11.5	Roots of a polynomial	139
11.6	Exercises	145
12	Linear Algebra – Simultaneous Equations	147
12.1	Introduction	147
12.2	Gaussian elimination	147
12.3	Iterative methods	154
12.4	Exercises	162
13	Characters and User-defined Scalar Types	164
13.1	Characters	164
13.2	Scalar types	166
13.3	Character graphics	168
13.4	Exercises	174
14	Data Summary	176
14.1	Introduction	176
14.2	Frequency distributions and histograms	176
14.3	Moments	188
14.4	Exercises	192
15	Random Numbers	195
15.1	Random and pseudo-random numbers	195
15.2	Tests for randomness	198
15.3	Random numbers from other distributions	201
15.4	Exercises	205
16	Records, Complex Numbers and the Fourier Transform	207
16.1	Introduction	207
16.2	Complex arithmetic	209
16.3	Discrete Fourier transform	210
16.4	Exercises	216
17	Pointers and Polynomials	217
17.1	Introduction	217
17.2	Pointer types	218

17.3	Linked structures	219
17.4	Pointers and procedures	225
17.5	Polynomial arithmetic	226
17.6	Exercises	236
18	Simulating Queues	237
18.1	Introduction	237
18.2	Simple queues	237
18.3	Doubly linked lists and rings	242
18.4	Exercises	251
19	Sets	254
19.1	Introduction	254
19.2	Assignment to sets	255
19.3	Set operations	256
19.4	Programming with sets	258
19.5	Binary relationships	260
19.6	Exercise	263
20	Trees and Graphs	264
20.1	Introduction	264
20.2	Undirected graphs	265
20.3	Trees	270
20.4	Directed graphs	277
20.5	Exercises	281
<i>Appendix A: Standard Functions and Procedures</i>		283
<i>Appendix B: Reserved and Pre-defined Words</i>		285
<i>References</i>		286
<i>Index</i>		287

Preface

Pascal is a rich and versatile computer language, designed to allow construction of programs that closely relate to the algorithm proposed as the solution to a particular problem. Well-written Pascal programs, therefore, are easy to read, understand and modify.

It is this feature of the language which makes it attractive as a first programming language. It encourages the programmer to carefully break problems into component parts which can then be translated into the programming language. As a result, it is best learnt alongside the process of learning how to solve problems numerically since the features of a problem's solution often have corresponding constructions in the language. Conversely, the problems supply suitable material for purposeful practice in using the computer.

This book has evolved out of a course on programming for first-year mathematicians at Leeds University. As well as teaching programming there was an additional requirement that the students be introduced to numerical problems related to their other courses. This performs the dual role of motivating those students who have not realised how important it is for the modern mathematician to be able to program, while allowing them to do numerical exercises related to material covered in their other courses.

The course, and subsequently the book, therefore were designed around the need to program numerical problems. The development of programming knowledge is dictated by the problems and no feature of the language, with the possible exceptions of procedures and functions, is introduced until it is required by the areas of application.

Thus, as well as introducing the Pascal language, this book is intended to serve as an introduction to solving numerical problems that arise in a first-year university Mathematics course. These problems are presented from a first principle point of view, with detailed proofs and justifications usually omitted.

The first part of the book (chapters 2-13) covers the material that I have at some stage included in the twelve lecture course given to our new Mathematics students at Leeds. It should however be stressed that not all the application areas have been covered in a single run of the course. This section covers the ideas of numerical methods such as iteration, integration, first-order differential equations and solving simultaneous equations. It is based solely on the use of INTEGER, REAL and BOOLEAN scalar types and ARRAYS, together with the constructions which allow looping and choice of alternative paths through a program.

The second part of the book (chapters 14–20) explores briefly the special features of Pascal such as **RECORDs**, **POINTERS** and **SETs**, again relating them to problems of interest to mathematicians. Thus complex numbers and Fourier transforms illustrate the use of **RECORDs**, and polynomial manipulation and queues are used to motivate and illustrate **POINTERS**. Finally, **SETs** and **POINTERS** are applied to examples involving trees and graphs.

Exercises are provided at the end of most chapters. Most of these are designed to give practice on the ideas developed in the preceding chapter, while others are intended to explore mathematical ideas numerically.

I would like to acknowledge the many helpful comments, suggestions and corrections made by colleagues and students. In particular, I would like to thank Patrick Constable, Geoffrey James, Allan Watkins, David Knapp, David Salinger, Tim Pullan and Stephen Mobbs who have read through various parts of the manuscript at some stage in its development. I would also like to express my appreciation of three independent reviewers for their criticisms, corrections and suggestions which I have found very useful. Naturally, any errors that remain are my responsibility.

Finally I would like to thank my wife Jane for her help and encouragement while I was writing the book.