

Introduction to Programming with Fortran

Ian Chivers · Jane Sleightholme

Introduction to Programming with Fortran

Fourth Edition

 Springer

Ian Chivers
Rhymney Consulting
London
UK

Jane Sleightholme
Fortranplus
London
UK

ISBN 978-3-319-75501-4 ISBN 978-3-319-75502-1 (eBook)
<https://doi.org/10.1007/978-3-319-75502-1>

Library of Congress Control Number: 2018942915

1st and 2nd edition: © Springer-Verlag London Limited 2006, 2012

3rd edition: © Springer International Publishing Switzerland 2015

4th edition: © Springer Nature Switzerland AG 2018, corrected publication 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG
part of Springer Nature

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

The Yorkshire connection dedicates the book to Steve, Mark and Jonathan. The Welsh connection dedicates the book to Joan, Martin and Jenny.

Acknowledgements

The material in the book has evolved first

- from our combined experience of working in Computing Services within the University of London at
 - King’s College, IDC (1986–2002) and JS (1985–2008)
 - Chelsea College, JS (1978–1985)
 - Imperial College, IDC (1978–1986)

in the teaching, advice and support of Fortran and related areas, and second

- in the provision of commercial training courses. The following are some of the organisations we’ve provided training for:
 - AMEC, Warrington
 - Aveva, Cambridge
 - AWE, Aldermaston
 - Centre for Ecology and Hydrology, Wallingford
 - DTU—Danish Technical University
 - Environment Agency, Worthing
 - Esso Petroleum, Fawley
 - JET—Joint European Torus
 - The Met Office, Bracknell and Exeter
 - National Nuclear Laboratory
 - Natural Resources Canada, Ottawa
 - Petroleum Geo-Services (PGS), Houston and Weybridge
 - QinetiQ, Farnborough
 - RAF Waddington
 - Ricardo Software
 - Risk Management Solutions
 - Rolls Royce, Derby
 - SHMU, Slovak Hydrometeorological Institute, Bratislava, Slovakia
 - University of Ulster, Jordanstown, Northern Ireland

- VCS, Germany
- Veritas DGC Ltd., Crawley
- Westland Helicopters, Yeovil.

Thanks are due to:

- The staff and students at King’s College, Chelsea College and Imperial College.
- The people who have attended the commercial courses. It has been great fun teaching you and things have been very lively at times.
- The people on the Fortran 90 list and comp.lang.fortran. Access to the expertise of several hundred people involved in the use and development of Fortran on a daily basis across a wide range of disciplines is inestimable.
- The people at NAG for the provision of beta test versions of their Fortran compilers and technical help and support.
- The people at Intel for the provision of beta test versions of their Fortran compilers and technical help and support.
- The people running the Archer service for their help.
- The people at Oracle who helped with the C Interop examples.
- The staff and facilities at PTR Associates. It is a pleasure training there.
- Helmut Michels at the Max Planck Institute for permission to use the *dislin* library.
- The patience of our families during the time required to develop the courses upon which this book is based and whilst preparing the camera-ready copy.
- Helen Desmond and Nancy Wade-Jones at Springer for their enthusiasm and encouragement when things were going wrong!

Our Fortran home page is:

<https://www.fortranplus.co.uk/>

All of the programme examples can be found there.

If you would like to contact us, our email addresses are:

Ian Chivers: ian@rhymneyconsulting.co.uk

Jane Sleightholme: jane@fortranplus.co.uk

The manuscript was produced using Springer’s L^AT_EX style sheet. We used proT_EXt, MiK_TE_X and T_EXnicCentre on the Windows platform. The graphs and plots were produced using the *dislin* graphics library. We started using T_EX at Imperial College on several CDC systems in the 1980s. T_EX and L^AT_EX have come a long way since then and plain T_EX seems a distant memory.

Contents

1	Overview	1
1.1	Introduction	1
1.2	Program Examples	4
1.3	Web Addresses	5
2	Introduction to Problem Solving	7
2.1	Introduction	8
2.2	Natural Language	8
2.3	Artificial Language	8
2.3.1	Notations	9
2.4	Resume	9
2.5	Algorithms	9
2.5.1	Top-Down	10
2.5.2	Bottom-Up	10
2.5.3	Stepwise Refinement	11
2.6	Modular Programming	11
2.7	Object Oriented Programming	12
2.8	Systems Analysis and Design	12
2.8.1	Problem Definition	12
2.8.2	Feasibility Study and Fact Finding	13
2.8.3	Analysis	13
2.8.4	Design	13
2.8.5	Detailed Design	13
2.8.6	Implementation	13
2.8.7	Evaluation and Testing	14
2.8.8	Maintenance	14
2.9	Unified Modelling Language - UML	14
2.10	Conclusions	15
2.11	Problems	15
2.12	Bibliography	15

3	Introduction to Programming Languages	19
3.1	Introduction	19
3.2	Some Early Theoretical Work	20
3.3	What Is a Programming Language?	20
3.4	Program Language Development and Engineering	20
3.5	The Early Days	20
3.5.1	Fortran's Origins	21
3.5.2	Fortran 77	21
3.5.3	Cobol	22
3.5.4	Algol	22
3.6	Chomsky and Program Language Development	23
3.7	Lisp	23
3.8	Snobol	24
3.9	Second-Generation Languages	24
3.9.1	PL/I and Algol 68	24
3.9.2	Simula	24
3.9.3	Pascal	25
3.9.4	APL	25
3.9.5	Basic	25
3.9.6	C	26
3.10	Some Other Strands in Language Development	27
3.10.1	Abstraction, Stepwise Refinement and Modules	27
3.10.2	Structured Programming	27
3.10.3	Data Structuring and Procedural Programming	27
3.10.4	Standardisation	28
3.11	Ada	28
3.12	Modula	29
3.13	Modula 2	29
3.14	Other Language Developments	30
3.14.1	Logo	30
3.14.2	Postscript, \TeX and \LaTeX	30
3.14.3	Prolog	31
3.14.4	SQL	31
3.14.5	ICON	31
3.15	Object Oriented Programming	32
3.15.1	Simula	32
3.15.2	Smalltalk	32
3.15.3	Oberon and Oberon 2	33
3.15.4	Eiffel	33
3.15.5	C++	34
3.15.6	Java	35
3.15.7	C#	36
3.15.8	Python	36

- 3.16 Back to Fortran! 38
 - 3.16.1 Fortran 90 38
 - 3.16.2 Fortran 95 38
 - 3.16.3 ISO Technical Reports TR15580 and TR15581 39
 - 3.16.4 Fortran 2003 39
 - 3.16.5 DTR 19767 Enhanced Module Facilities 41
 - 3.16.6 Fortran 2008 41
 - 3.16.7 TS 29113 Further Interoperability of Fortran
with C 43
 - 3.16.8 Fortran 2018 43
- 3.17 Fortran Discussion Lists 46
- 3.18 ACM Fortran Forum 46
- 3.19 Other Sources 47
- 3.20 Summary 47
- 3.21 Bibliography 47
- 4 Introduction to Programming 55**
 - 4.1 Introduction 55
 - 4.2 Language Strengths and Weaknesses 56
 - 4.3 Elements of a Programming Language 56
 - 4.3.1 Data Description Statements 57
 - 4.3.2 Control Structures 57
 - 4.3.3 Data-Processing Statements 57
 - 4.3.4 Input and Output (I/O) Statements 57
 - 4.4 Example 1: Simple Text I/O 58
 - 4.5 Variables — Name, Type and Value 59
 - 4.6 Example 2: Simple Numeric I/O and Arithmetic 60
 - 4.7 Some More Fortran Rules 61
 - 4.8 Fortran Character Set 62
 - 4.9 Good Programming Guidelines 64
 - 4.10 Compilers Used 65
 - 4.11 Compiler Documentation 66
 - 4.11.1 gfortran 66
 - 4.11.2 IBM 67
 - 4.11.3 Intel 67
 - 4.11.4 Nag 68
 - 4.11.5 Oracle/Sun 68
 - 4.12 Program Development 69
 - 4.13 Problems 70
- 5 Arithmetic 71**
 - 5.1 Introduction 72
 - 5.2 The Fortran Operators and the Arithmetic Assignment
Statement 72

5.3	Example 1: Simple Arithmetic Expressions in Fortran	73
5.4	The Fortran Rules for Arithmetic	74
5.5	Expression Equivalence	76
5.6	Rounding and Truncation	77
5.7	Example 2: Type Conversion and Assignment	77
5.8	Example 3: Integer Division and Real Assignment	78
5.9	Example 4: Time Taken for Light to Travel from the Sun to Earth	79
5.10	The Parameter Attribute	80
5.11	Round Off Errors and Computer Arithmetic	81
5.12	Relative and Absolute Errors	82
5.13	Example 5: Relative and Absolute Error	83
5.14	Range, Precision and Size of Numbers	84
5.15	Overflow and Underflow	85
	5.15.1 Example 6: Overflow	85
	5.15.2 Example 7: Underflow	86
5.16	Health Warning: Optional Reading, Beginners Are Advised to Leave Until Later	87
	5.16.1 Positional Number Systems	87
	5.16.2 Fortran Representational Models	87
5.17	Kind Types	89
	5.17.1 Example 8: Testing What Kind Types Are Available	89
5.18	Testing the Numerical Representation of Different Kind Types on a System	90
5.19	Example 9: Using the Numeric Inquiry Functions with Integer Types	91
5.20	Example 10: Using the Numeric Inquiry Functions with Real Types	93
5.21	gfortran Support for Intel Extended (80 bit) Precision	97
5.22	Example 11: Literal Real Constants in a Calculation	97
5.23	Summation and Finite Precision	98
	5.23.1 Example 12: Rounding Problem	99
5.24	Example 13: Binary Representation of Different Integer Kind Type Numbers	99
5.25	Example 14: Binary Representation of a Real Number	101
5.26	Example 15: Initialisation of Physical Constants, Version 1	102
5.27	Example 16: Initialisation of Physical Constants, Version 2	104
5.28	Example 17: Initialisation of Physical Constants, Version 3	104
5.29	Summary of How to Select the Appropriate Kind Type	105

- 5.30 Variable Status 106
- 5.31 Fortran and the IEEE 754 Standard 106
- 5.32 Summary 107
- 5.33 Bibliography 107
- 5.34 Problems 108
- 6 Arrays 1: Some Fundamentals 113**
 - 6.1 Tables of Data 113
 - 6.1.1 Telephone Directory 114
 - 6.1.2 Book Catalogue 114
 - 6.1.3 Examination Marks or Results 115
 - 6.1.4 Monthly Rainfall 115
 - 6.2 Arrays in Fortran 116
 - 6.2.1 The Dimension Attribute 116
 - 6.2.2 An Index 116
 - 6.2.3 Control Structure 116
 - 6.3 Example 1: Monthly Rainfall 117
 - 6.4 Possible Missing Data 119
 - 6.5 Example 2: People’s Weights and Setting the Array
Size With a Parameter 121
 - 6.6 Summary 122
 - 6.7 Problems 123
- 7 Arrays 2: Further Examples 127**
 - 7.1 Varying the Array Size at Run Time 128
 - 7.1.1 Example 1: Allocatable Arrays 128
 - 7.2 Higher-Dimension Arrays 129
 - 7.2.1 Example 2: Two Dimensional Arrays
and a Map 129
 - 7.2.2 Example 3: Sensible Tabular Output 131
 - 7.2.3 Example 4: Average of Three Sets of Values 132
 - 7.2.4 Example 5: Booking Arrangements in a Theatre
or Cinema 133
 - 7.3 Additional Forms of the Dimension Attribute and Do Loop
Statement 134
 - 7.3.1 Example 6: Voltage from -20 to +20 Volts 134
 - 7.3.2 Example 7: Longitude from -180 to +180 135
 - 7.3.3 Notes 136
 - 7.4 The Do Loop and Straight Repetition 136
 - 7.4.1 Example 8: Table of Liquid Conversion
Measurements 136
 - 7.4.2 Example 9: Means and Standard Deviations 137

7.5	Summary	138
7.6	Problems	139
8	Whole Array and Additional Array Features	143
8.1	Terminology	143
8.2	Array Element Ordering	144
8.3	Whole Array Manipulation	144
8.4	Assignment	144
8.5	Expressions	145
8.6	Example 1: Rank 1 Whole Arrays in Fortran	146
8.7	Example 2: Rank 2 Whole Arrays in Fortran	147
8.8	Array Sections	148
8.8.1	Example 3: Rank 1 Array Sections	148
8.8.2	Example 4: Rank 2 Array Sections	149
8.9	Array Constructors	150
8.9.1	Example 5: Rank 1 Array Initialisation — Explicit Values	150
8.9.2	Example 6: Rank 1 Array Initialisation Using an Implied Do Loop	151
8.9.3	Example 7: Rank 1 Arrays and the <code>dot_product</code> Intrinsic	152
8.9.4	Initialising Rank 2 Arrays	152
8.9.5	Example 8: Initialising a Rank 2 Array	153
8.10	Miscellaneous Array Examples	154
8.10.1	Example 9: Rank 1 Arrays and a Stride of 2	154
8.10.2	Example 10: Rank 1 Array and the Sum Intrinsic Function	155
8.10.3	Example 11: Rank 2 Arrays and the Sum Intrinsic Function	156
8.10.4	Example 12: Masked Array Assignment and the <code>where</code> Statement	157
8.10.5	Notes	158
8.11	Array Element Ordering in More Detail	159
8.11.1	Example 13: Array Element Ordering	159
8.12	Physical and Virtual Memory	160
8.13	Type Declaration Statement Summary	161
8.14	Summary	161
8.15	Problems	162
8.16	Bibliography	162
8.16.1	DEC Alpha	162
8.16.2	AMD	162
8.16.3	Intel	163

9	Output of Results	165
9.1	Introduction	166
9.2	Integers and the <code>i</code> Format or Edit Descriptor	166
9.2.1	Example 1: Twelve Times Table	166
9.2.2	Example 2: Integer Overflow and the <code>i</code> Edit Descriptor	167
9.3	Reals and the <code>f</code> Edit Descriptor	168
9.3.1	Example 3: Imperial Pints and US Pints	168
9.3.2	Example 4: Imperial Pints and Litres	169
9.3.3	Example 5: Narrow Field Widths and the <code>f</code> Edit Descriptor	170
9.3.4	Example 6: Overflow and the <code>f</code> Edit Descriptor	171
9.4	Reals and the <code>e</code> Edit Descriptor	171
9.4.1	Example 7: Simple <code>e</code> Edit Descriptor Example	172
9.5	Reals and the <code>g</code> Edit Descriptor	173
9.5.1	Example 8: Simple <code>g</code> Edit Descriptor Example	173
9.6	Spaces	174
9.6.1	Example 9: Three Ways of Generating Spaces	174
9.7	Characters — a Format or Edit Descriptor	175
9.7.1	Example 10: Character Output and the Edit Descriptor	175
9.7.2	Example 11: Character, Integer and Real Output in a Format Statement	176
9.8	Common Mistakes	177
9.9	Files in Fortran	177
9.9.1	The <code>open</code> Statement	177
9.9.2	The <code>close</code> Statement	178
9.9.3	Example 12: Open and Close Usage	178
9.9.4	Example 13: Timing of Writing Formatted Files	179
9.9.5	Example 14: Timing of Writing Unformatted Files	181
9.10	Example 15: Implied Do Loops and Array Sections for Array Output	182
9.11	Example 16: Repetition and Whole Array Output	184
9.12	Example 17: Choosing the Decimal Symbol	184
9.13	Example 18: Alternative Format Specification Using a String	185
9.14	Example 19: Alternative Format Specification Using a Character Variable	186
9.15	The Remaining Control and Data Edit Descriptors	186
9.16	Summary	187
9.17	Problems	188

- 10 Reading in Data** 191
 - 10.1 Reading from Files 191
 - 10.2 Example 1: Reading Integer Data 192
 - 10.3 Example 2: Reading Real Data 194
 - 10.4 Met Office Historic Station Data 195
 - 10.5 Example 3: Reading One Column of Data from a File 196
 - 10.6 Example 4: Skipping Lines in a File 198
 - 10.7 Example 5: Reading from Several Files Consecutively 199
 - 10.8 Example 6: Reading Using Array Sections 201
 - 10.9 Example 7: Reading Using Internal Files 203
 - 10.10 Example 8: Timing of Reading Formatted Files 204
 - 10.11 Example 9: Timing of Reading Unformatted Files 205
 - 10.12 Summary 206
 - 10.13 Problems 207
- 11 Summary of I/O Concepts** 209
 - 11.1 I/O Concepts and Statements 209
 - 11.2 Records 210
 - 11.3 File Access 210
 - 11.4 The open Statement 210
 - 11.5 Data Transfer Statements 211
 - 11.6 The inquire Statement 212
 - 11.7 Error, End of Record and End of File 213
 - 11.7.1 Error Conditions and the err= Specifier 214
 - 11.7.2 End-of-File Condition and the end= Specifier 214
 - 11.7.3 End-of-Record Condition and the eor= Specifier 214
 - 11.7.4 iostat= Specifier 214
 - 11.7.5 iomsg= Specifier 215
 - 11.8 Examples 215
 - 11.8.1 Example 1: Simple Use of the read, write, open, close, unit Features 215
 - 11.8.2 Example 2: Using iostat to Test for Errors 216
 - 11.8.3 Example 3: Use of newunit and lentrin 217
 - 11.9 Unit Numbering 219
 - 11.10 Summary 219
 - 11.11 Problems 219
- 12 Functions** 221
 - 12.1 Introduction 221
 - 12.2 An Introduction to Predefined Functions and Their Use 222
 - 12.2.1 Example 1: Simple Function Usage 223
 - 12.3 Generic Functions 224
 - 12.3.1 Example 2: The abs Generic Function 224

- 12.4 Elemental Functions 224
 - 12.4.1 Example 3: Elemental Function Use 224
- 12.5 Transformational Functions 225
 - 12.5.1 Example 4: Simple Transformational Use 225
 - 12.5.2 Example 5: Intrinsic `dot_product` Use 225
- 12.6 Notes on Function Usage 226
- 12.7 Example 6: Easter 226
- 12.8 Intrinsic Procedures 228
- 12.9 Supplying Your Own Functions 229
 - 12.9.1 Example 7: Simple User Defined Function 229
- 12.10 An Introduction to the Scope of Variables, Local Variables and Interface Checking 231
- 12.11 Recursive Functions 232
 - 12.11.1 Example 8: Recursive Factorial Evaluation 232
- 12.12 Example 9: Recursive Version of `gcd` 234
- 12.13 Example 10: `gcd` After Removing Recursion 235
- 12.14 Internal Functions 236
 - 12.14.1 Example 11: Stirling’s Approximation 236
- 12.15 Pure Functions 237
 - 12.15.1 Pure Constraints 238
- 12.16 Elemental Functions 238
- 12.17 Resume 239
- 12.18 Formal Syntax 239
- 12.19 Rules and Restrictions 240
- 12.20 Problems 241
- 12.21 Bibliography 241
 - 12.21.1 Recursion and Problem Solving 242
- 13 Control Structures and Execution Control 243**
 - 13.1 Introduction 244
 - 13.2 Selection Among Courses of Action 244
 - 13.3 The Block If Statement 245
 - 13.3.1 Example 1: Quadratic Roots 247
 - 13.3.2 Example 2: Date Calculation 248
 - 13.4 The Case Statement 249
 - 13.4.1 Example 3: Simple Calculator 250
 - 13.4.2 Example 4: Counting Vowels, Consonants, etc. 251
 - 13.5 The Various Forms of the Do Statement 252
 - 13.5.1 Example 5: Sentinel Usage 253
 - 13.5.2 Cycle and Exit 254

- 13.5.3 Example 6: The Evaluation of `e**x` 254
- 13.5.4 Example 7: Wave Breaking on an Offshore Reef . . . 256
- 13.6 Do Concurrent 257
- 13.7 Summary 260
 - 13.7.1 Control Structure Formal Syntax 260
- 13.8 Problems 261
- 13.9 Bibliography 263
- 14 Characters 265**
 - 14.1 Introduction 265
 - 14.2 Character Input 266
 - 14.2.1 Example 1: The `*` Edit Descriptor 267
 - 14.2.2 Example 2: The `a` Edit Descriptor 267
 - 14.3 Character Operators 268
 - 14.4 Character Substrings 269
 - 14.4.1 Example 3: Stripping Blanks from a String 270
 - 14.5 Character Functions 270
 - 14.5.1 Example 4: The `index` Character Function 270
 - 14.5.2 The `len` and `len_trim` Functions 271
 - 14.5.3 Example 5: Using `len` and `len_trim` 271
 - 14.6 Collating Sequence 272
 - 14.7 Example 6: Finding Out About the Character Set Available 273
 - 14.8 The `scan` Function 275
 - 14.8.1 Example 7: Using the `scan` Function 275
 - 14.9 Summary 276
 - 14.10 Problems 277
- 15 Complex 281**
 - 15.1 Introduction 281
 - 15.2 Example 1: Use of `cmplx`, `aimag` and `conjg` 282
 - 15.3 Example 2: Polar Coordinate Example 283
 - 15.4 Complex and Kind Type 284
 - 15.5 Summary 284
 - 15.6 Problem 284
- 16 Logical 287**
 - 16.1 Introduction 287
 - 16.2 I/O 290
 - 16.3 Summary 291
 - 16.4 Problems 291
- 17 Introduction to Derived Types 293**
 - 17.1 Introduction 293
 - 17.2 Example 1: Dates 294

- 17.3 Type Definition 295
- 17.4 Variable Definition 295
 - 17.4.1 Example 2: Variant of Example 1 Using
Modules 295
- 17.5 Example 3: Address Lists 296
- 17.6 Example 4: Nested User Defined Types 298
- 17.7 Problem 300
- 18 An Introduction to Pointers 301**
 - 18.1 Introduction 301
 - 18.2 Example 1: Illustrating Some Basic Pointer Concepts 302
 - 18.3 Example 2: The associated Intrinsic Function 304
 - 18.4 Example 3: Referencing Pointer Variables Before
Allocation or Pointer Assignment 304
 - 18.4.1 gfortran 305
 - 18.4.2 Intel 305
 - 18.4.3 Nag 306
 - 18.5 Example 4: Pointer Allocation and Assignment 306
 - 18.6 Memory Leak Examples 307
 - 18.6.1 Example 5: Simple Memory Leak 307
 - 18.6.2 Example 6: More Memory Leaks 308
 - 18.7 Non-standard Pointer Example 309
 - 18.7.1 Example 7: Using the C loc Function 309
 - 18.8 Problems 310
- 19 Introduction to Subroutines 313**
 - 19.1 Introduction 313
 - 19.2 Example 1: Roots of a Quadratic Equation 314
 - 19.2.1 Referencing a Subroutine 316
 - 19.2.2 Dummy Arguments or Parameters and Actual
Arguments 317
 - 19.2.3 The intent Attribute 317
 - 19.2.4 Local Variables 317
 - 19.2.5 Local Variables and the save Attribute 317
 - 19.2.6 Scope of Variables 318
 - 19.2.7 Status of the Action Carried Out
in the Subroutine 318
 - 19.2.8 Modules ‘Containing’ Procedures 318
 - 19.3 Why Bother with Subroutines? 319
 - 19.4 Summary 319
 - 19.5 Problems 320

20	Subroutines: 2	321
20.1	More on Parameter Passing	321
20.1.1	Assumed-Shape Array	321
20.1.2	Deferred-Shape Array	322
20.1.3	Automatic Arrays	322
20.1.4	Allocatable Dummy Arrays	322
20.1.5	Keyword and Optional Arguments	322
20.2	Example 1: Assumed Shape Parameter Passing	323
20.2.1	Notes	325
20.3	Example 2: Character Arguments and Assumed-Length Dummy Arguments	325
20.4	Example 3: Rank 2 and Higher Arrays as Parameters	327
20.4.1	Notes	328
20.5	Example 4: Automatic Arrays and Median Calculation	329
20.5.1	Internal Subroutines and Scope	332
20.6	Example 5: Recursive Subroutines – Quicksort	332
20.6.1	Note — Recursive Subroutine	335
20.6.2	Note — Flexible Design	335
20.7	Example 6: Allocatable Dummy Arrays	336
20.8	Example 7: Elemental Subroutines	339
20.9	Summary	340
20.10	Problems	340
20.11	Bibliography	342
20.12	Commercial Numerical and Statistical Subroutine Libraries	342
21	Modules	343
21.1	Introduction	343
21.2	Basic Module Syntax	344
21.3	Modules for Global Data	344
21.4	Example 1: Modules for Precision Specification and Constant Definition	345
21.5	Example 2: Modules for Globally Sharing Data	347
21.6	Modules for Derived Data Types	349
21.7	Example 3: Person Data Type	350
21.8	Example 4: A Module for Simple Timing of a Program	352
21.9	private, public and protected Attributes	353
21.10	The use Statement	354
21.11	Notes on Module Usage and Compilation	354
21.12	Example 5: Modules and Include Statements	354
21.13	Formal Syntax	355
21.13.1	Interface	355
21.13.2	Implicit and Explicit Interfaces	356
21.13.3	Explicit Interface	356

21.14	Summary	356
21.15	Problems	357
22	Data Structuring in Fortran	359
22.1	Introduction	359
22.2	Example 1: Singly Linked List: Reading an Unknown Amount of Text	359
22.3	Example 2: Reading in an Arbitrary Number of Reals Using a Linked List and Copying to an Array	363
22.4	Example 3: Ragged Arrays	365
22.5	Example 4: Ragged Arrays and Variable Sized Data Sets	366
22.6	Example 5: Perfectly Balanced Tree	369
22.7	Example 6: Date Class	372
22.7.1	Notes: DST in the USA	383
22.8	Example 7: Date Data Type with USA and ISO Support	383
22.9	Bibliography	388
22.10	Problems	389
23	An Introduction to Algorithms and the Big O Notation	391
23.1	Introduction	391
23.2	Basic Background	392
23.3	Brief Explanation	392
23.4	Example 1: Order Calculations	393
23.5	Sorting	394
23.6	Basic Array and Linked List Performance	395
23.7	Bibliography	395
24	Operator Overloading	397
24.1	Introduction	397
24.2	Other Languages	397
24.3	Example 1: Overloading the Addition (+) Operator	398
24.4	Problem	399
25	Generic Programming	401
25.1	Introduction	401
25.2	Generic Programming and Other Languages	401
25.3	Example 1: Sorting Reals and Integers	402
25.3.1	Generic Quicksort in C++	410
25.3.2	Generic Quicksort in C#	411
25.4	Example 2: Generic Statistics Module	413
25.5	Problems	419
25.6	Bibliography	420
25.6.1	Generic Programming References	420
25.6.2	Generic Programming and C++	420
25.6.3	Generic Programming and C#	420

26	Mathematical and Numerical Examples	421
26.1	Introduction	422
26.2	Example 1: Using Linked Lists for Sparse Matrix Problems	422
26.2.1	Inner Product of Two Sparse Vectors	422
26.3	Example 2: Solving a System of First-Order Ordinary Differential Equations Using Runge–Kutta–Merson	427
26.3.1	Note: Alternative Form of the Allocate Statement	434
26.3.2	Note: Automatic Arrays	434
26.3.3	Note: Subroutine as a Dummy Procedure Argument:	435
26.3.4	Note: Compilation When Using Modules	435
26.3.5	Keyword and Optional Argument Variation	436
26.4	Example 3: A Subroutine to Extract the Diagonal Elements of a Matrix	437
26.5	Example 4: The Solution of Linear Equations Using Gaussian Elimination	438
26.5.1	Notes	443
26.6	Example 5: Allocatable Function Results	444
26.7	Example 6: Elemental e^{**x} Function	445
26.8	Example 7: Absolute and Relative Errors Involved in Subtraction Using 32 bit Reals	447
26.9	Example 8: Absolute and Relative Errors Involved in Subtraction Using 64 bit Reals	448
26.10	Problems	450
26.11	Bibliography	450
27	Parameterised Derived Types (PDTs) in Fortran	453
27.1	Introduction	453
27.2	Example 1: Linked List Parameterised by Real Kind	454
27.3	Example 2: Ragged Array Parameterised by Real Kind Type	456
27.4	Example 3: Specifying <code>len</code> in a PDT	457
27.5	Problems	460
28	Introduction to Object Oriented Programming	461
28.1	Introduction	461
28.2	Brief Review of the History of Object Oriented Programming	461
28.3	Background Technical Material	462
28.3.1	The Concept of Type	463
28.3.2	Type Classification	463
28.3.3	Set of Values	463

28.3.4	Type	463
28.3.5	Class	463
28.3.6	Attributes	464
28.3.7	Passed Object Dummy Arguments	464
28.3.8	Derived Types and Structure Constructors	465
28.3.9	Structure Constructors and Generic Names	465
28.3.10	Assignment	466
28.3.11	Intrinsic Assignment Statement	466
28.3.12	Defined Assignment Statement	466
28.3.13	Polymorphic Variables	467
28.3.14	Executable Constructs Containing Blocks	467
28.3.15	The <code>associate</code> Construct	467
28.3.16	The <code>select type</code> Construct	467
28.4	Example 1: The Basic Shape Class	468
28.4.1	Key Points	470
28.4.2	Notes	473
28.5	Example 2: Base Class with Private Data	473
28.6	Example 3: Using an Interface to Use the Class Name for the Structure Constructor	475
28.6.1	Public and Private Accessibility	477
28.7	Example 4: Simple Inheritance	478
28.7.1	Base Shape Class	478
28.7.2	Circle - Derived Type 1	478
28.7.3	Rectangle - Derived Type 2	481
28.7.4	Simple Inheritance Test Program	483
28.8	Example 5: Polymorphism and Dynamic Binding	485
28.8.1	Base Shape Class	485
28.8.2	Circle - Derived Type 1	487
28.8.3	Rectangle - Derived Type 2	488
28.8.4	Shape Wrapper Module	488
28.8.5	Display Subroutine	488
28.8.6	Test Program for Polymorphism and Dynamic Binding	489
28.9	Fortran 2008 and Polymorphic Intrinsic Assignment	492
28.10	Summary	494
28.11	Problems	494
28.12	Further Reading	495
29	Additional Object Oriented Examples	497
29.1	Introduction	497
29.2	The Date Class	498
29.3	Example 1: The Base Date Class	498
29.3.1	Day and Month Name Module	500

29.3.2	Date Module	500
29.3.3	Diff Output Between Original Module and New oo Module	507
29.3.4	Main Program	515
29.3.5	Diff Output Between Original Program and New oo Test Program	518
29.4	Example 2: Simple Inheritance Based on an ISO Date Format	523
29.4.1	ISO Date Module	523
29.4.2	ISO Test Program	527
29.5	Example 3: Using the Two Date Formats and Showing Polymorphism and Dynamic Binding	531
29.5.1	Date Wrapper Module	531
29.5.2	Polymorphic and Dynamic Binding Test Program	532
29.6	Dates, Date Validity and Calendars	533
29.6.1	Calendars	533
29.6.2	Date Formats	534
29.6.3	Other Calendar Systems	535
29.6.4	Proleptic Gregorian Calendar	535
29.6.5	References	535
29.7	An Abstract Base Class in Fortran	536
29.8	Problems	540
29.9	Bibliography	542
30	Introduction to Submodules	543
30.1	Introduction	543
30.2	Brief Technical Background	544
30.3	Example 1: Rewrite of the Date Class Using Submodules	546
30.4	Example 2: Rewrite of the First Order RKM ODE Solver Using Modules	560
30.5	Problems	566
30.6	Bibliography	566
31	Introduction to Parallel Programming	567
31.1	Introduction	567
31.2	Parallel Computing Classification	569
31.3	Amdahl's Law	569
31.3.1	Amdahl's Law Graph 1–8 Processors or Cores	570
31.3.2	Amdahl's Law Graph 2–64 Processors or Cores	570
31.4	Gustafson's Law	571
31.4.1	Gustafson's Law Graph 1–64 Processors or Cores	571
31.5	Memory Access	571

31.6	Cache	572
31.7	Bandwidth and Latency	572
31.8	Flynn’s Taxonomy	573
31.9	Consistency Models	573
31.10	Threads and Threading	573
31.11	Threads and Processes	574
31.12	Data Dependencies	574
31.13	Race Conditions	574
31.14	Mutual Exclusion - Mutex	574
31.15	Monitors	574
31.16	Locks	574
31.17	Synchronization	575
31.18	Granularity and Types of Parallelism	575
31.19	Partitioned Global Address Space - PGAS	575
31.20	Fortran and Parallel Programming	575
31.21	MPI	576
31.22	OpenMP	577
31.23	Coarray Fortran	578
31.24	Other Parallel Options	578
	31.24.1 PVM	579
	31.24.2 HPF	579
31.25	Top 500 Supercomputers	579
31.26	Summary	580
32	MPI - Message Passing Interface	581
32.1	Introduction	581
32.2	MPI Programming	581
32.3	Compiler and Implementation Combination	582
32.4	Individual Implementation	582
	32.4.1 MPICH2	582
	32.4.2 Open MPI	583
32.5	Compiler and MPI Combinations Used in the Book	583
	32.5.1 Cray Archer System	583
32.6	The MPI Memory Model	584
32.7	Example 1: Hello World	584
32.8	Example 2: Hello World Using Send and Receive	586
32.9	Example 3: Serial Solution for pi Calculation	590
32.10	Example 4: Parallel Solution for pi Calculation	594
32.11	Example 5: Work Sharing Between Processes	600
32.12	Summary	604
32.13	Problem	604

- 33 OpenMP** 605
 - 33.1 Introduction 605
 - 33.2 OpenMP Memory Model 606
 - 33.3 Example 1: Hello World 607
 - 33.4 Example 2: Hello World Using Default Variable Data Scoping 610
 - 33.5 Example 3: Hello World with Private `thread_number` variable 612
 - 33.6 Example 4: Parallel Solution for pi Calculation 613
 - 33.7 Example 5: Comparing the Timing of Whole Array Syntax, Simple Do Loops, Do Concurrent and an OpenMP Solution 616
 - 33.8 Summary 619
 - 33.9 Problem 619
- 34 Coarray Fortran** 621
 - 34.1 Introduction 621
 - 34.2 Some Basic Coarray Terminology 622
 - 34.3 Example 1: Hello World 623
 - 34.4 Example 2: Broadcasting Data 624
 - 34.5 Example 3: Parallel Solution for pi Calculation 625
 - 34.6 Example 4: Work Sharing 628
 - 34.7 Summary 632
 - 34.8 Problem 632
- 35 C Interop** 633
 - 35.1 Introduction 633
 - 35.2 The `iso_c_binding` Module 633
 - 35.3 Named Constants and Derived Types in the Module 634
 - 35.4 Character Interoperability 635
 - 35.5 Procedures in the Module 635
 - 35.6 Interoperability of Intrinsic Types 635
 - 35.7 Other Aspects of Interoperability 636
 - 35.7.1 Interoperability with C Pointer Types 636
 - 35.7.2 Interoperability of Scalar Variables 636
 - 35.7.3 Interoperability of Array Variables 636
 - 35.7.4 Interoperability of Procedures and Procedure Interfaces 636
 - 35.7.5 Interoperation with C Global Variables 636
 - 35.7.6 Binding Labels for Common Blocks and Variables 637
 - 35.7.7 Interoperation with C Functions 637
 - 35.8 Compilers Used in the Examples 637
 - 35.9 Example 1: Kind Type Support 638

- 35.10 Example 2: Fortran Calling a C Function 640
- 35.11 Example 3: C Calling a Fortran Function 641
- 35.12 Example 4: C++ Calling a Fortran Function 642
- 35.13 Example 5: Passing an Array from Fortran to C 643
- 35.14 Example 6: Passing an Array from C to Fortran 644
- 35.15 Example 7: Passing an Array from C++ to Fortran 645
- 35.16 Example 8: Passing a Rank 2 Array from Fortran to C 646
- 35.17 Example 9: Passing a Rank 2 Array from C to Fortran 647
- 35.18 Example 10: Passing a Rank 2 Array from C++ to Fortran 649
- 35.19 Example 11: Passing a Rank 2 Array from C++ to Fortran and Taking Care of Array Storage 651
 - 35.19.1 Compiler Switches 653
- 35.20 Example 12: Passing a Rank 2 Array from C to Fortran and Taking Care of Array Storage 654
 - 35.20.1 Compiler Switches 656
- 35.21 Example 13: Passing a Fortran Character Variable to C 657
- 35.22 Example 14: Passing a Fortran Character Variable to C++ 660
- 35.23 c_loc Examples on Our Web Site 662
 - 35.23.1 c_loc(x) Description 662
- 35.24 Problem 664
- 36 IEEE Arithmetic 665**
 - 36.1 Introduction 665
 - 36.2 History 665
 - 36.3 IEEE Specifications 667
 - 36.4 Floating Point Formats 669
 - 36.5 Procedure Summary 669
 - 36.6 General Comments About the Standard 671
 - 36.7 Resume 672
 - 36.8 Fortran Support for IEEE Arithmetic 673
 - 36.9 Derived Types and Constants Defined in the Modules 673
 - 36.9.1 ieee_exceptions 673
 - 36.9.2 ieee_arithmetic 674
 - 36.9.3 ieee_features 675
 - 36.9.4 Further Information 675
 - 36.10 Example 1: Testing IEEE Support 675
 - 36.11 Example 2: Testing What Flags Are Supported 676
 - 36.12 Example 3: Overflow 678
 - 36.13 Example 4: Underflow 678
 - 36.14 Example 5: Inexact Summation 679
 - 36.15 Example 6: NAN and Other Specials 682
 - 36.16 Summary 683
 - 36.17 Bibliography 684

- 36.17.1 Web-Based Sources 684
- 36.17.2 Hardware Sources 685
- 36.17.3 Operating Systems 686
- 36.18 Problem 687
- 37 Derived Type I/O 689**
 - 37.1 Introduction 689
 - 37.2 User-Defined Derived-Type Editing 691
 - 37.3 Example 1: Basic Syntax, No Parameters in Call 691
 - 37.4 Example 2: Extended Syntax, Passing Parameters 694
 - 37.5 Example 3: Basic Syntax with Timing 699
 - 37.6 Example 4: Extended Syntax with Timing 701
 - 37.7 Summary 702
 - 37.8 Problem 702
- 38 Sorting and Searching 703**
 - 38.1 Example 1: Generic Recursive Quicksort Example
with Timing Details 703
 - 38.2 Example 2: Non Recursive Quicksort Example
with Timing Details 708
 - 38.2.1 Notes - Version Control Systems 714
 - 38.3 Subroutine and Function Libraries 715
 - 38.4 The Nag Library for SMP and Multicore 715
 - 38.5 Example 3: Calling the Nag m01caf Sorting Routine 715
 - 38.6 Example 4: Sorting an Array of a Derived Type 718
 - 38.6.1 Compare Function 719
 - 38.6.2 Fortran Sources 719
 - 38.6.3 Date Module 719
 - 38.6.4 Sort Module 725
 - 38.6.5 Main Program 726
 - 38.7 Example 5: Binary Search Example 728
 - 38.8 Problems 732
- 39 Handling Missing Data in Statistics Calculations 733**
 - 39.1 Introduction 733
 - 39.2 Example 1: Program to Download and Save the Data
Files Locally 733
 - 39.3 Example 2: The Sed Script and Command File That
Converts the Missing Values 736
 - 39.4 Example 3: The Program to Do the Statistics Calculations 737
 - 39.5 Example 4: Met Office Utility Program 746
 - 39.6 Bibliography 751
 - 39.7 Problem 752

40 Converting from Fortran 77 753

40.1 Introduction 753

40.2 Deleted Features from Fortran 90 754

40.3 Deleted Features from Fortran 2008 755

40.4 Obsolescent Features 755

40.4.1 Alternate Return 756

40.4.2 Computed GO TO Statement 756

40.4.3 Statement Functions 756

40.4.4 DATA Statements Among Executables 756

40.4.5 Assumed Character Length Functions 757

40.4.6 Fixed Form Source 757

40.4.7 CHARACTER* Form of CHARACTER Declaration 757

40.4.8 ENTRY Statements 757

40.4.9 Label DO Statement 758

40.4.10 COMMON and EQUIVALENCE Statements and the Block Data Program Unit 758

40.4.11 Specific Names for Intrinsic Functions 758

40.4.12 FORALL Construct and Statement 758

40.5 Better Alternatives 758

40.6 Free and Commercial Conversion Tools 759

40.6.1 Convert 760

40.6.2 Forcheck 760

40.6.3 Nag Compiler Polish Tool 760

40.6.4 Plusfort 761

40.7 Example 1: Using the plusFORT Tool Suite from Polyhedron Software 761

40.7.1 Original Fortran 66 761

40.7.2 Fortran 77 Version 762

40.7.3 Fortran 90 Version 762

40.8 Example 2: Leaving as Fortran 77 763

40.9 Example 3: Simple Conversion to Fortran 90 764

40.10 Example 4: Simple Syntax Conversion to Modern Fortran 769

40.11 Example 5: Date Case Study 775

40.12 Example 6: Creating 64 Bit Integer and 128 Bit Real Sorting Subroutines from the Netlib Sorting Routines 784

40.13 Summary 796

40.14 Problems 796

41 Graphics Libraries - Simple Dislin Usage 797

41.1 Introduction 797

41.2 The Dislin Graphics Library 797

41.3 Example 1: Using Dislin to Plot Amdahl’s Law Graph 1–8 Processors or Cores 798

- 41.4 Example 2: Using Dislin to Plot Amdahl’s Law Graph 2–64
Processors or Cores 800
- 41.5 Example 3: Using Dislin to Plot Gustafson’s Law Graph
1–64 Processors or Cores 802
- 41.6 Example 4: Using Dislin to Plot Tsunami Events 804
- 41.7 Example 5: Using Dislin to Plot the Met Office Data 811
- 41.8 Graphics Production Notes 814
- 41.9 Bibliography 814
- 41.10 Problems 815
- 42 Abstract Interfaces and Procedure Pointers 817**
 - 42.1 Introduction 817
 - 42.2 Example 1: Abstract Interfaces and Procedure Pointers 817
 - 42.3 Problem 820
- Correction to: Characters C1**
- Appendix A: Glossary 821**
- Appendix B: Attribute Declarations and Specifications 833**
- Appendix C: Compatibility 835**
- Appendix D: Intrinsic Functions and Procedures 841**
- Appendix E: Text Extracts, English, Latin and coded 911**
- Appendix F: Formal syntax 913**
- Appendix G: Compiler Options 917**
- Index 933**

List of Tables

Table 3.1	C++ standardisation history	34
Table 3.2	Fortran standardisation history	45
Table 4.1	Variable name, type and value	60
Table 4.2	The Fortran character set	63
Table 4.3	ASCII character set	63
Table 5.1	Fortran operators	72
Table 5.2	Hierarchy or precedence of the Fortran operators	75
Table 5.3	Some commonly used physical constants	81
Table 5.4	Word size and integer numbers	84
Table 5.5	Word size and real numbers	84
Table 5.6	Kind inquiry functions	90
Table 5.7	Numeric inquiry functions	91
Table 5.8	Integer kind type parameter name and integer value	92
Table 5.9	Integer kind and huge comparison	93
Table 5.10	Extended real type comparison	96
Table 6.1	Fortran statement ordering	123
Table 8.1	Array element ordering in Fortran	159
Table 9.1	Summary of data edit descriptors	187
Table 9.2	Text edit descriptors	187
Table 11.1	Open statement options	211
Table 11.2	Data transfer statement options	212
Table 11.3	Inquire statement options	212
Table 12.1	Some of the intrinsic functions available in Fortran	223
Table 13.1	Fortran logical and relational operators	246
Table 14.1	String functions in Fortran	277
Table 16.1	Simple truth table	289
Table 16.2	Fortran operator hierarchy	289
Table 23.1	Big O notation and complexity	392
Table 23.2	Quicksort and insertion sort comparison	395
Table 23.3	Array and linked list performance	395
Table 25.1	ch2502 results	419

Table 28.1	Polymorphic intrinsic assignment support.	494
Table 31.1	Bandwidth and latency.	572
Table 32.1	Intel I7 with hyperthreading.	599
Table 35.1	iso_c_binding module - named constants.	634
Table 35.2	C Interop character interoperability	635
Table 35.3	Compilers used.	637
Table 35.4	Basic C Interop table	639
Table 36.1	Computer hardware and manufacturers.	666
Table 36.2	Operating systems	666
Table 36.3	IEEE formats	669
Table 36.4	IEEE Arithmetic module procedure summary.	670
Table 36.5	IEEE Exceptions module procedure summary	671
Table 36.6	Compiler IEEE support for various precisions	676
Table 38.1	Generic recursive quicksort timing.	707
Table 38.2	Non recursive quicksort timing.	714
Table 38.3	Sixty four bit real sort timings	717
Table 38.4	Nag sort m01caf timing	717
Table C.1	Previous editions of the Fortran standard	835
Table D.1	Argument and return type abbreviations.	841
Table D.2	Classes of function	842
Table D.3	Common optional arguments	842
Table D.4	Standard generic intrinsic procedure summary	845
Table D.5	Intrinsic functions by standard - Fortran 90 to Fortran 2018	903