

Contents

1	Overview	1
1.1	Introduction	1
1.2	Program Examples	4
1.3	Web Addresses	4
2	Introduction to Problem Solving	5
2.1	Introduction	6
2.2	Natural Language	6
2.3	Artificial Language	7
2.3.1	Notations	7
2.4	Resume	7
2.5	Algorithms	8
2.5.1	Top-Down	8
2.5.2	Bottom-Up	8
2.5.3	Stepwise Refinement	9
2.6	Modular Programming	10
2.7	Object Oriented Programming	10
2.8	Systems Analysis and Design	10
2.8.1	Problem Definition	10
2.8.2	Feasibility Study and Fact Finding	11
2.8.3	Analysis	11
2.8.4	Design	11
2.8.5	Detailed Design	11
2.8.6	Implementation	12
2.8.7	Evaluation and Testing	12
2.8.8	Maintenance	12
2.9	Unified Modeling Language—UML	13
2.10	Conclusions	13
2.11	Problems	14

3	Introduction to Programming Languages	17
3.1	Introduction	17
3.2	Some Early Theoretical Work	17
3.3	What Is a Programming Language?	18
3.4	Program Language Development and Engineering	18
3.5	The Early Days	18
3.5.1	Fortran's Origins	18
3.5.2	Fortran 77	19
3.5.3	Cobol	19
3.5.4	Algol	20
3.6	Chomsky and Program Language Development	20
3.7	Lisp	21
3.8	Snobol	21
3.9	Second-Generation Languages	22
3.9.1	PL/1 and Algol 68	22
3.9.2	Simula	22
3.9.3	Pascal	22
3.9.4	APL	23
3.9.5	Basic	23
3.9.6	C	23
3.10	Some Other Strands in Language Development	24
3.10.1	Abstraction, Stepwise Refinement and Modules	24
3.10.2	Structured Programming	24
3.10.3	Data Structuring and Procedural Programming	25
3.10.4	Standardisation	25
3.11	Ada	26
3.12	Modula	26
3.13	Modula 2	27
3.14	Other Language Developments	27
3.14.1	Logo	27
3.14.2	Postscript, TEX and L ^A T _E X	28
3.14.3	Prolog	28
3.14.4	SQL	28
3.14.5	ICON	29
3.15	Object Oriented Programming	29
3.15.1	Simula	30
3.15.2	Smalltalk	30
3.15.3	Oberon and Oberon 2	30
3.15.4	Eiffel	31
3.15.5	C++	31
3.15.6	Java	32
3.15.7	C#	33

3.16	Back to Fortran!	34
3.16.1	Fortran 90	34
3.16.2	Fortran 95	35
3.16.3	ISO Technical Reports TR15580 and TR15581	35
3.16.4	Fortran 2003	36
3.16.5	DTR 19767 Enhanced Module Facilities	37
3.16.6	Fortran 2008	37
3.16.7	TS 29113 Further Interoperability of Fortran with C	37
3.16.8	Fortran 2015	37
3.17	Fortran Discussion Lists	38
3.18	ACM Fortran Forum	39
3.19	Other Sources	39
3.20	Summary	39
4	Introduction to Programming	47
4.1	Introduction	47
4.2	Language Strengths and Weaknesses	48
4.3	Elements of a Programming Language	48
4.3.1	Data Description Statements	49
4.3.2	Control Structures	49
4.3.3	Data-Processing Statements	49
4.3.4	Input and Output (I/O) Statements	49
4.4	Example 1: Simple Text I/O	50
4.5	Variables—Name, Type and Value	51
4.6	Example 2: Simple Numeric I/O and Arithmetic	52
4.7	Some More Fortran Rules	54
4.8	Fortran Character Set	54
4.9	Good Programming Guidelines	55
4.10	Compilers Used	57
4.11	Compiler Documentation	58
4.11.1	Gfortran	58
4.11.2	IBM	58
4.11.3	Intel	59
4.11.4	Nag	60
4.11.5	Oracle/Sun	60
4.12	Program Development	60
4.13	Reference Text for the Fortran 2003 Standard	61
4.14	Problems	62
5	Arithmetic	63
5.1	Introduction	64
5.2	The Fortran Operators and the Arithmetic Assignment Statement	64

5.3	Example 1: Simple Arithmetic Expressions in Fortran	65
5.4	The Fortran Rules for Arithmetic	66
5.5	Expression Equivalence	68
5.6	Rounding and Truncation	69
5.7	Example 2: Type Conversion and Assignment	69
5.8	Example 3: Integer Division and Real Assignment	70
5.9	Example 4: Time Taken for Light to Travel from the Sun to Earth	71
5.10	The Parameter Attribute	72
5.11	Round Off Errors and Computer Arithmetic	73
5.12	Relative and Absolute Errors	74
5.13	Example 5: Relative and Absolute Error.	75
5.14	Range, Precision and Size of Numbers.	76
5.15	Overflow and Underflow	77
	5.15.1 Example 6: Overflow	77
	5.15.2 Example 7: Underflow	78
5.16	Health Warning: Optional Reading, Beginners Are Advised to Leave Until Later	79
	5.16.1 Positional Number Systems.	79
	5.16.2 Fortran Representational Models	79
5.17	Testing the Numerical Representation of Different Kind Types on a System	81
5.18	Example 8: Using the Numeric Inquiry Functions with Integer Types	82
5.19	Example 9: Using the Numeric Inquiry Functions with Real Types	84
5.20	Example 10: Literal Real Constants in a Calculation	87
5.21	Summation and Finite Precision	88
	5.21.1 Example 11: Rounding Problem	89
5.22	Example 12: Binary Representation of Different Integer Kind Type Numbers	89
5.23	Example 13: Binary Representation of a Real Number.	91
5.24	Example 14: Initialisation of Physical Constants, Version 1.	92
5.25	Example 15: Initialisation of Physical Constants, Version 2.	94
5.26	Example 16: Initialisation of Physical Constants, Version 3.	94
5.27	Summary of How to Select the Appropriate Kind Type	95
5.28	Variable Status	96
5.29	Fortran and the IEEE 754 Standard	96
5.30	Summary	97
5.31	Problems	98

6	Arrays 1: Some Fundamentals	103
6.1	Tables of Data	103
6.1.1	Telephone Directory	104
6.1.2	Book Catalogue.	104
6.1.3	Examination Marks or Results.	104
6.1.4	Monthly Rainfall	105
6.2	Arrays in Fortran.	106
6.2.1	The Dimension Attribute	106
6.2.2	An Index	106
6.2.3	Control Structure	106
6.3	Example 1: Monthly Rainfall	107
6.4	Possible Missing Data	109
6.5	Example 2: People's Weights and Setting the Array Size With a Parameter	110
6.6	Summary	112
6.7	Problems	113
7	Arrays 2: Further Examples	115
7.1	Varying the Array Size at Run Time	115
7.1.1	Example 1: Allocatable Arrays	116
7.2	Higher-Dimension Arrays.	117
7.2.1	Example 2: Two Dimensional Arrays and a Map.	117
7.2.2	Example 3: Sensible Tabular Output	119
7.2.3	Example 4: Average of Three Sets of Values	120
7.2.4	Example 5: Booking Arrangements in a Theatre or Cinema	121
7.3	Additional Forms of the Dimension Attribute and Do Loop Statement	122
7.3.1	Example 6: Voltage from -20 to +20 Volts	122
7.3.2	Example 7: Longitude from -180 to +180	123
7.3.3	Notes	123
7.4	The Do Loop and Straight Repetition	124
7.4.1	Example 8: Table of Liquid Conversion Measurements	124
7.4.2	Example 9: Means and Standard Deviations	124
7.5	Summary	125
7.6	Problems	127
8	Whole Array and Additional Array Features	131
8.1	Terminology	131
8.2	Array Element Ordering	132
8.3	Whole Array Manipulation	132
8.4	Assignment	132
8.5	Expressions	133

8.6	Example 1: Rank 1 Whole Arrays in Fortran	134
8.7	Example 2: Rank 2 Whole Arrays in Fortran	135
8.8	Array Sections	136
8.8.1	Example 3: Rank 1 Array Sections	136
8.8.2	Example 4: Rank 2 Array Sections	136
8.9	Array Constructors	138
8.9.1	Example 5: Rank 1 Array Initialisation—Explicit Values	138
8.9.2	Example 6: Rank 1 Array Initialisation Using An Implied do loop	139
8.9.3	Example 7: Rank 1 Arrays and the <code>dot_product</code> Intrinsic	140
8.9.4	Initialising Rank 2 Arrays.	140
8.9.5	Example 8: Initialising a Rank 2 Array	141
8.10	Miscellaneous Array Examples	142
8.10.1	Example 9: Rank 1 Arrays and a Stride of 2.	142
8.10.2	Example 10: Rank 1 Array and the <code>sum</code> Intrinsic Function.	143
8.10.3	Example 11: Rank 2 Arrays and the <code>sum</code> Intrinsic Function.	144
8.10.4	Example 12: Masked Array Assignment and the <code>where</code> Statement.	145
8.10.5	Notes.	146
8.11	The <code>forall</code> Statement and <code>forall</code> Construct.	146
8.11.1	Syntax	146
8.12	Array Element Ordering in More Detail.	147
8.12.1	Example 13: Array Element Ordering	147
8.13	Physical and Virtual Memory	148
8.14	Type Declaration Statement Summary	149
8.15	Summary	149
8.16	Problems	150
9	Output of Results	153
9.1	Introduction	153
9.2	Integers and the <code>i</code> Format or Edit Descriptor	154
9.2.1	Example 1: Twelve Times Table	154
9.2.2	Example 2: Integer Overflow and the <code>i</code> Edit Descriptor.	155
9.3	Reals and the <code>f</code> Edit Descriptor	156
9.3.1	Example 3: Imperial Pints and US Pints.	156
9.3.2	Example 4: Imperial Pints and Litres	157
9.3.3	Example 5: Narrow Field Widths and the <code>f</code> Edit Descriptor	158
9.3.4	Example 6: Overflow and the <code>f</code> Edit Descriptor	158

9.4	Reals and the <code>e</code> Edit Descriptor	159
9.4.1	Example 7: Simple <code>e</code> Edit Descriptor Example	159
9.5	Reals and the <code>g</code> Edit Descriptor	161
9.5.1	Example 8: Simple <code>g</code> Edit Descriptor Example	161
9.6	Spaces	162
9.6.1	Example 9: Three Ways of Generating Spaces	162
9.7	Characters—a Format or Edit Descriptor	163
9.7.1	Example 10: Character Output and the <code>a</code> Edit Descriptor.	163
9.7.2	Example 11: Character, Integer and Real Output in a Format Statement	164
9.8	Common Mistakes.	164
9.9	Files in Fortran	165
9.9.1	The <code>open</code> Statement	165
9.9.2	The <code>close</code> Statement	166
9.9.3	Example 12: Open and Close Usage	166
9.9.4	Example 13: Timing of Writing Formatted Files	167
9.9.5	Example 14: Timing of Writing Unformatted Files	169
9.10	Example 15: Implied Do Loops and Array Sections for Array Output.	170
9.11	Example 16: Repetition and Whole Array Output	171
9.12	Example 17: Choosing the Decimal Symbol	172
9.13	Example 18: Alternative Format Specification Using a String.	173
9.14	Example 19: Alternative Format Specification Using a Character Variable.	174
9.15	The Remaining Control and Data Edit Descriptors.	174
9.16	Summary	174
9.17	Problems	175
10	Reading in Data	179
10.1	Reading from Files	179
10.2	Example 1: Reading Integer Data	179
10.3	Example 2: Reading Real Data	182
10.4	Met Office Historic Station Data	183
10.5	Example 3: Reading One Column of Data from a File	184
10.6	Example 4: Skipping Lines in a File	186
10.7	Example 5: Reading from Several Files Consecutively	187
10.8	Example 6: Reading Using Array Sections	189
10.9	Example 7: Reading Using Internal Files	190
10.10	Example 8: Timing of Reading Formatted Files	192
10.11	Example 9: Timing of Reading Unformatted Files.	193
10.12	Summary	194
10.13	Problems	194

11	Summary of I/O Concepts	197
11.1	I/O Concepts and Statements	197
11.2	Records	198
11.3	File Access	198
11.4	The <code>open</code> Statement	198
11.5	Data Transfer Statements	200
11.6	The <code>inquire</code> Statement	201
11.7	Error, End of Record and End of File	202
11.7.1	Error Conditions and the <code>err=</code> Specifier	202
11.7.2	End-of-file Condition and the <code>end=</code> Specifier	202
11.7.3	End-of-Record Condition and the <code>eor=</code> Specifier	202
11.7.4	<code>iostat=</code> Specifier	203
11.7.5	<code>iomsg=</code> Specifier	203
11.8	Examples	203
11.8.1	Example 1: Simple Use of the <code>read</code> , <code>write</code> , <code>open</code> , <code>close</code> , <code>unit</code> Features	203
11.8.2	Example 2: Using <code>iostat</code> to Test for Errors	204
11.9	Example 3: Use of <code>newunit</code> and <code>lentrin</code>	205
11.10	Unit Numbering	207
11.11	Summary	207
11.12	Problems	207
12	Functions	209
12.1	Introduction	209
12.2	An Introduction to Predefined Functions and Their Use	210
12.2.1	Example 1: Simple Function Usage	210
12.3	Generic Functions	211
12.3.1	Example 2: The <code>abs</code> Generic Function	211
12.4	Elemental Functions	212
12.4.1	Example 3: Elemental Function Use	212
12.5	Transformational Functions	212
12.5.1	Example 4: Simple Transformational Use	212
12.5.2	Example 5: Intrinsic <code>dot_product</code> Use	213
12.6	Notes on Function Usage	213
12.7	Example 6: <code>Easter</code>	214
12.8	Intrinsic Procedures	216
12.9	Supplying Your Own Functions	216
12.9.1	Example 7: Simple User Defined Function	216
12.10	An Introduction to the Scope of Variables, Local Variables and Interface Checking	219
12.11	Recursive Functions	220
12.11.1	Example 8: Recursive Factorial Evaluation	220
12.12	Example 9: Recursive Version of GCD	222
12.13	Example 10: GCD After Removing Recursion	223

12.14	Internal Functions	223
12.14.1	Example 11: Stirling's Approximation	224
12.15	Pure Functions	224
12.15.1	Pure Constraints	225
12.16	Elemental Functions	226
12.17	Resume	227
12.18	Formal Syntax	227
12.19	Rules and Restrictions	228
12.20	Problems	229
13	Control Structures	231
13.1	Introduction	231
13.2	Selection Among Courses of Action	232
13.3	The Block if Statement	232
13.3.1	Example 1: Quadratic Roots	235
13.3.2	Example 2: Date Calculation.	236
13.4	The Case Statement.	237
13.4.1	Example 3: Simple Calculator.	237
13.4.2	Example 4: Counting Vowels, Consonants, etc.	238
13.5	The Three Forms of the do Statement	239
13.5.1	Example 5: Sentinel Usage	240
13.5.2	Cycle and Exit	241
13.5.3	Example 6: The Evaluation of e^*x	242
13.5.4	Example 7: Wave Breaking on an Offshore Reef.	243
13.6	Summary	245
13.6.1	Control Structure Formal Syntax	245
13.7	Problems	246
14	Characters	249
14.1	Introduction	249
14.2	Character Input	250
14.2.1	Example 1: The * Edit Descriptor	251
14.2.2	Example 2: The a Edit Descriptor.	251
14.3	Character Operators.	251
14.4	Character Substrings	253
14.4.1	Example 3: Stripping Blanks from a String.	253
14.5	Character Functions.	254
14.5.1	Example 4: The index Character Function	254
14.5.2	The len and len_trim Functions	255
14.5.3	Example 5: Using len and len_trim.	255
14.6	Collating Sequence	256
14.7	Example 6: Finding Out About the Character Set Available.	257
14.8	The scan Function.	258
14.8.1	Example 7: Using the scan Function	258

14.9	Summary	260
14.10	Problems	260
15	Complex.	265
15.1	Introduction	265
15.2	Example 1: Use of <code>cmplx</code> , <code>aimag</code> and <code>conjg</code>	266
15.3	Example 2: Polar Coordinate Example	266
15.4	Complex and Kind Type	268
15.5	Summary	268
15.6	Problems	268
16	Logical.	269
16.1	Introduction	269
16.2	I/O	272
16.3	Summary	273
16.4	Problems	273
17	Introduction to Derived Types.	275
17.1	Introduction	275
17.2	Example 1: Dates	276
17.3	Type Definition	277
17.4	Variable Definition	277
17.4.1	Example 2: Variant of Example 1 Using Modules ...	277
17.5	Example 3: Address Lists	278
17.6	Example 4: Nested User Defined Types	280
17.7	Problems	282
18	An Introduction to Pointers.	283
18.1	Introduction	283
18.2	Example 1: Illustrating Some Basic Pointer Concepts	284
18.3	Example 2: The <code>associated</code> Intrinsic Function	285
18.4	Example 3: Referencing Pointer Variables Before Allocation or Pointer Assignment	286
18.4.1	<code>gfortran</code>	287
18.4.2	<code>Intel</code>	287
18.4.3	<code>Nag</code>	288
18.5	Example 4: Pointer Allocation and Assignment	288
18.6	Memory Leak Examples	288
18.6.1	Example 5: Simple Memory Leak	289
18.6.2	Example 6: More Memory Leaks	289
18.7	Non-standard Pointer Example	290
18.7.1	Example 7: Using the <code>C loc</code> Function	291
18.8	Problems	292

19	Introduction to Subroutines.	293
19.1	Introduction	293
19.2	Example 1: Roots of a Quadratic Equation	294
19.2.1	Referencing a Subroutine	296
19.2.2	Dummy Arguments or Parameters and Actual Arguments	296
19.2.3	The intent Attribute	297
19.2.4	Local Variables	297
19.2.5	Local Variables and the save Attribute	297
19.2.6	Scope of Variables	298
19.2.7	Status of the Action Carried Out in the Subroutine	298
19.2.8	Modules 'Containing' Procedures	298
19.3	Why Bother with Subroutines?	299
19.4	Summary	299
19.5	Problems	299
20	Subroutines: 2	301
20.1	More on Parameter Passing	301
20.1.1	Assumed-Shape Array	301
20.1.2	Deferred-Shape Array	301
20.1.3	Automatic Arrays	302
20.1.4	Allocatable Dummy Arrays	302
20.1.5	Keyword and Optional Arguments	302
20.2	Example 1: Assumed Shape Parameter Passing	303
20.2.1	Notes	305
20.3	Example 2: Character Arguments and Assumed-Length Dummy Arguments	305
20.4	Example 3: Rank 2 and Higher Arrays as Parameters	306
20.4.1	Notes	308
20.5	Example 4: Automatic Arrays and Median Calculation	308
20.5.1	Internal Subroutines and Scope	311
20.6	Example 5: Recursive Subroutines—Quicksort	312
20.6.1	Note—Recursive Subroutine	315
20.6.2	Note—Flexible Design	315
20.7	Example 6: Allocatable Dummy Arrays	315
20.8	Example 7: Elemental Subroutines	318
20.9	Summary	319
20.10	Problems	320
20.11	Commercial Numerical and Statistical Subroutine Libraries	322
21	Modules	323
21.1	Introduction	323
21.2	Basic Module Syntax	324
21.3	Modules for Global Data	324

21.4	Example 1: Modules for Precision Specification and Constant Definition	325
21.5	Example 2: Modules for Globally Sharing Data	327
21.6	Modules for Derived Data Types.	329
21.7	Example 3: Person Data Type.	329
21.8	Example 4: A Module for Simple Timing of a Program	332
21.9	private, public and protected attributes.	333
21.10	The use statement	333
21.11	Notes on Module Usage and Compilation.	334
21.12	Formal Syntax	334
	21.12.1 Interface.	334
	21.12.2 Implicit and Explicit Interfaces	334
	21.12.3 Explicit Interface	335
21.13	Summary	335
21.14	Problems	336
22	Data Structuring in Fortran	337
22.1	Introduction	337
22.2	Example 1: Singly Linked List: Reading an Unknown Amount of Text	337
22.3	Example 2: Reading in an Arbitrary Number of Reals Using a Linked List and Copying to an Array	341
22.4	Example 3: Ragged Arrays.	342
22.5	Example 4: Ragged Arrays and Variable Sized Data Sets. . .	343
22.6	Example 5: Perfectly Balanced Tree	344
22.7	Example 6: Date class	346
	22.7.1 Notes: DST in the USA	357
22.8	Problems	357
23	An Introduction to Algorithms and the Big O Notation	359
23.1	Introduction	359
23.2	Basic Background	360
23.3	Brief Explanation	360
23.4	Example 1	361
23.5	Sorting.	362
23.6	Basic Array and Linked List Performance	363
24	Operator Overloading.	365
24.1	Introduction	365
24.2	Other Languages	365
24.3	Example 1: Overloading the Addition (+) Operator.	366
24.4	Problems	367

25	Generic Programming	369
25.1	Introduction	369
25.2	Generic Programming and Other Languages	369
25.3	Example 1: Sorting Reals and Integers	370
25.3.1	Generic Quicksort in C++	377
25.3.2	Generic Quicksort in C#	379
25.4	Example 2: Generic Statistics Module	380
25.5	Problems	386
26	Mathematical Examples	387
26.1	Introduction	387
26.2	Example 1: Using Linked Lists for Sparse Matrix Problems	387
26.2.1	Inner Product of Two Sparse Vectors	388
26.3	Example 2: Solving a System of First-Order Ordinary Differential Equations Using Runge-Kutta-Merson	392
26.3.1	Note: Alternative Form of the Allocate Statement	399
26.3.2	Note: Automatic Arrays	399
26.3.3	Note: Subroutine as a Dummy Procedure Argument	400
26.3.4	Note: Compilation When Using Modules	400
26.3.5	Keyword and Optional Argument Variation	400
26.4	Example 3: A Subroutine to Extract the Diagonal Elements of a Matrix	402
26.5	Example 4: The Solution of Linear Equations Using Gaussian Elimination	403
26.5.1	Notes	407
26.6	Example 5: Allocatable Function Results	408
26.7	Example 6: Elemental e**x Function	410
26.8	Problems	411
27	Parameterised Derived Types (PDTs) in Fortran	413
27.1	Introduction	413
27.2	Example 1: Linked List Parameterised by Real Kind	414
27.3	Example 2: Ragged Array Parameterised by Real Kind Type	416
27.4	Example 3: Specifying len in a PDT	418
27.5	Problems	420
28	Object Oriented Programming	421
28.1	Introduction	421
28.2	Brief Review of the History of Object Oriented Programming	421
28.3	Background Technical Material	422

28.4	Type Declaration Statements	423
28.4.1	Type	423
28.4.2	Class	423
28.4.3	Attributes	423
28.4.4	Passed Object Dummy Arguments	424
28.4.5	Derived Types and Structure Constructors	425
28.4.6	Structure Constructors and Generic Names	425
28.4.7	Assignment	425
28.4.8	Intrinsic Assignment Statement	426
28.4.9	Defined Assignment Statement	426
28.4.10	Polymorphic Variables	426
28.4.11	Executable Constructs Containing Blocks	426
28.4.12	The <code>associate</code> Construct	426
28.4.13	The <code>select type</code> Construct	427
28.5	Example 1: The Basic Shape Class	427
28.5.1	Key Points	429
28.5.2	Notes	432
28.5.3	Example 2: Base Class with Private Data	432
28.5.4	Example 3: Using an Interface to Use the Class Name for the Structure Constructor	433
28.5.5	Public and Private Accessibility	435
28.6	Example 4: Simple Inheritance	436
28.6.1	Base Shape Class	436
28.6.2	Circle—Derived Type 1	436
28.6.3	Rectangle—Derived Type 2	439
28.6.4	Simple Inheritance Test Program	441
28.7	Example 5: Polymorphism and Dynamic Binding	443
28.7.1	Base Shape Class	443
28.7.2	Circle—Derived Type 1	445
28.7.3	Rectangle—Derived Type 2	445
28.7.4	Shape Wrapper Module	445
28.7.5	Display Subroutine	445
28.7.6	Test Program for Polymorphism and Dynamic Binding	446
28.8	Summary	449
28.9	Problems	449
29	Introduction to Parallel Programming	451
29.1	Introduction	451
29.2	Parallel Computing Classification	452
29.3	Amdahl's Law	453
29.3.1	Amdahl's Law Graph 1—8 Processors or Cores	453
29.3.2	Amdahl's Law Graph 2—64 Processors or Cores	454

29.4	Gustafson's Law	454
29.4.1	Gustafson's Law Graph 1–64 Processors or Cores	455
29.5	Memory Access	455
29.6	Cache	455
29.7	Bandwidth and Latency	456
29.8	Flynn's Taxonomy	456
29.9	Consistency Models.	457
29.10	Threads and Threading.	457
29.11	Threads and Processes	457
29.12	Data Dependencies	457
29.13	Race Conditions	458
29.14	Mutual Exclusion-Mutex	458
29.15	Monitors	458
29.16	Locks	458
29.17	Synchronization	458
29.18	Granularity and Types of Parallelism.	459
29.19	Partitioned Global Address Space—PGAS	459
29.20	Fortran and Parallel Programming	459
29.21	MPI.	459
29.22	OpenMP	461
29.23	Coarray Fortran.	462
29.24	Other Parallel Options	462
29.24.1	PVM	463
29.24.2	HPF.	463
29.25	Top 500 Supercomputers	463
29.26	Summary	464
30	MPI—Message Passing Interface	465
30.1	Introduction	465
30.2	MPI Programming.	465
30.3	Compiler and Implementation Combination	466
30.4	Individual Implementation	466
30.4.1	MPICH2	466
30.4.2	Open MPI	466
30.5	Compiler and MPI Combinations Used in the Book	467
30.5.1	Cray Archer System.	467
30.6	The MPI Memory Model	467
30.7	Example: 1—Hello World	468
30.8	Example: 2—Hello World Using Send and Receive.	470
30.9	Example: 3—Serial Solution for Pi Calculation.	473
30.10	Example: 4—Parallel Solution for Pi Calculation.	478
30.11	Example: 5—Work Sharing Between Processes	483
30.12	Summary	487
30.13	Problems	487

31	OpenMP	489
31.1	Introduction	489
31.2	OpenMP Memory Model	490
31.3	Example 1: Hello World	491
31.4	Example 2: Hello World Using Default Variable Data Scoping	494
31.5	Example 3: Hello World with Private <code>thread_number</code> variable	495
31.6	Example 4: Parallel Solution for Pi Calculation	496
31.7	Summary	499
31.8	Problems	499
32	Coarray Fortran	501
32.1	Introduction	501
32.2	Coarray Terminology	502
32.3	Example 1: Hello World	503
32.4	Example 2: Broadcasting Data	504
32.5	Example 3: Parallel Solution for Pi Calculation	505
32.6	Example 4: Work Sharing	508
32.7	Summary	512
32.8	Problems	512
33	C Interop	513
33.1	Introduction	513
33.2	The <code>iso_c_binding</code> Module	513
33.3	Named Constants and Derived Types in the Module	513
33.4	Character Interoperability	514
33.5	Procedures in the Module	515
33.6	Interoperability of Intrinsic Types	515
33.7	Other Aspects of Interoperability	516
33.7.1	Interoperability with C Pointer Types	516
33.7.2	Interoperability of Scalar Variables	516
33.7.3	Interoperability of Array Variables	516
33.7.4	Interoperability of Procedures and Procedure Interfaces	516
33.7.5	Interoperation with C Global Variables	516
33.7.6	Binding Labels for Common Blocks and Variables	517
33.7.7	Interoperation with C Functions	517
33.8	Compilers Used in the Examples	517
33.9	Example 1: Kind Type Support	518
33.10	Example 2: Fortran Calling a C Function	520
33.11	Example 3: C Calling a Fortran Function	521
33.12	Example 4: C++ Calling a Fortran Function	522
33.13	Example 5: Passing an Array from Fortran to C	522
33.14	Example 6: Passing an Array from C to Fortran	524

33.15	Example 7: Passing an Array from C++ to Fortran	525
33.16	Example 8: Passing a Rank 2 Array from Fortran to C	526
33.17	Example 9: Passing a Rank 2 Array from C to Fortran	527
33.18	Example 10: Passing a Rank 2 Array from C++ to Fortran	528
33.19	Example 11: Passing a Rank 2 Array from C++ to Fortran and Taking Care of Array Storage.	530
33.19.1	Compiler Switches.	532
33.20	Example 12: Passing a Rank 2 Array from C to Fortran and Taking Care of Array Storage.	533
33.20.1	Compiler Switches.	535
33.21	Acknowledgements	535
33.22	c_loc Examples on our Web Site	535
33.22.1	c_loc(x) Description	536
33.23	Problems	537
34	IEEE Arithmetic.	539
34.1	Introduction	539
34.2	History	539
34.3	IEEE 754 Specifications.	541
34.3.1	Single Precision Floating Point Format.	542
34.3.2	Double Precision Floating Point Format	543
34.3.3	Two Classes of Extended Floating Point Formats.	543
34.3.4	Accuracy Requirements	544
34.3.5	Base Conversion—Converting Between Decimal and Binary Floating Point Formats and Vice Versa	544
34.3.6	Exception Handling	544
34.3.7	Rounding Directions	544
34.3.8	Rounding Precisions	545
34.4	Resume	545
34.5	Fortran Support for IEEE Arithmetic	545
34.6	Derived Types and Constants Defined in the Modules	546
34.6.1	ieee_exceptions	546
34.6.2	ieee_arithmetic	546
34.6.3	ieee_features	547
34.6.4	Further Information	547
34.7	Example 1: Testing IEEE Support.	548
34.8	Example 2: Testing What Flags are Supported	549
34.9	Example 3: Overflow.	550
34.10	Example 4: Underflow.	551
34.11	Example 5: Inexact Summation.	552
34.12	Example 6: NAN and Other Specials.	554
34.13	Summary	556
34.14	Problems	559

35	Miscellaneous Examples	561
35.1	Example 1: Date Data Type with USA and ISO Support	561
35.2	Example 2: Binary Search Example	566
35.3	Comparison of Three Sorting Methods	569
35.4	Example 3: Generic Recursive Quicksort Example with Timing Details	569
35.5	Example 4: Non Recursive Quicksort Example with Timing Details	572
35.5.1	Notes—Version Control Systems	577
35.6	Subroutine and Function Libraries	577
35.7	Numerical Libraries	578
35.8	Serial and Parallel Libraries	578
35.9	The Nag Library for SMP and Multicore	578
35.10	Example 5: Calling the Nag M01caf Sorting Routine	579
35.11	Graphics Libraries	581
35.12	The Dislin Graphics Library	581
35.12.1	Dislin Example 1: Amdahl's Law Graph 1–8 Processors or Cores	582
35.12.2	Dislin Example 2: Amdahl's Law Graph 2–64 Processors or Cores	583
35.12.3	Dislin Example 3: Gustafson's Law Graph 1–64 Processors or Cores	583
35.12.4	Dislin Example 4: The Display of Tsunami Events	583
35.13	Problems	589
36	Converting from Fortran 77	591
36.1	Introduction	591
36.2	Deleted Features	592
36.3	Obsolescent Features	592
36.3.1	Arithmetic If	592
36.3.2	Real and Double Precision Do Control Variables	592
36.3.3	Shared do Termination and Non-endo Termination	592
36.3.4	Alternate Return	592
36.3.5	Pause Statement	593
36.3.6	Assign and Assigned Goto Statements	593
36.3.7	Assigned Format Statements	593
36.3.8	H Editing	593
36.4	Better Alternatives	593
36.5	Commercial Conversion Tools	594
36.5.1	Convert	594
36.5.2	Forcheck	594
36.5.3	Forstruct	595

36.5.4	Forstudy	595
36.5.5	Plusfort	595
36.5.6	VAST/77to90	595
36.6	Example of PlusFORT Capability from Polyhedron Software	595
36.6.1	Original Fortran 66	596
36.6.2	Fortran 77 Version	596
36.6.3	Fortran 90 Version	597
36.7	Summary	598
Appendix A: Glossary		599
Appendix B: Intrinsic Functions and Procedures.		611
Appendix C: Text Extracts, English, Latin and coded		655
Appendix D: Formal Syntax		657
Appendix E: Compiler Options		661
Index		663