IAN CHIVERS AND JANE SLEIGHTHOLME

INTRODUCING FORTRAN 90

- Learn FORTRAN 90, the modern language of first choice for scientists, engineers and mathematicians
- Ideal for complete beginners with little programming knowledge
- Highlights the changes from FORTRAN 77 to FORTRAN 90
- Contains lots of examples to emphasise the language features

Springer

-370-1

2-005-370-1

Ian Chivers and Jane Sleightholme

INTRODUCING FORTRAN 90

جامعة البليدة
المكتبة المركزية

جامعة البليدة
المكتبة المركزية

Springer

1 Overview **1**
2 Introduction to Computer Systems **6**
2.1 The components of a computer system 7
2.2 Software 8
2.3 Problems 9
2.4 Bibliography 9
3 Introduction to Operating Systems **11**
3.1 History of Operating Systems 12
3.1.1 The 1940s 12
3.1.2 The 1950s 12
3.1.3 The 1960s 12
3.1.4 The 1960s and 1970s 12
3.1.5 The 1970s and 1980s 13
3.2 Networking 13
3.3 Problems 14
3.4 Bibliography 14
4 Introduction to Using a Computer System **15**
4.1 Files 16
4.2 Editors 16
4.3 Stand Alone Systems 16
4.4 Networked Systems 16
4.5 Multi-User Systems 17
4.6 Other Useful Things to Know 17
4.7 Bibliography 18
5 Introduction to Problem Solving **19**
5.1 Natural Language 20
5.2 Artificial Language 21
5.2.1 Notations 21
5.3 Resume 21
5.4 Algorithms 22
5.4.1 Top Down 22
5.4.2 Bottom up 22
5.4.3 Stepwise Refinement 23
5.5 Systems Analysis and Design 23
5.5.1 Problem Definition 24
5.5.2 Feasibility Study and Fact Finding 24
5.5.3 Analysis 24
5.5.4 Design 24
5.5.5 Detailed Design 24

5.5.6 Implementation 25
5.5.7 Evaluation and testing 25
5.5.8 Maintenance 25
5.6 Conclusions 25
5.7 Problems 26
5.8 Bibliography 26
6 Introduction to Programming Languages 28
6.1 Some Early Theoretical Work 29
6.2 What is a programming language ? 29
6.3 Program Language Development and Engineering 29
6.4 The Early Days 29
6.4.1 Fortran 30
6.4.2 Cobol 30
6.4.3 Algol 31
6.5 Chomsky and Program Language Development 31
6.6 Lisp 32
6.7 Snobol 32
6.8 Second Generation Languages 32
6.8.1 PL/1 and Algol 68 32
6.8.2 Simula 33
6.8.3 Pascal 33
6.8.4 APL 33
6.8.5 Basic 34
6.8.6 C 34
6.9 Some Other Strands in Language Development 34
6.9.1 Abstraction, Stepwise Refinement and Modules 34
6.9.2 Structured Programming 34
6.9.3 Standardisation 35
6.10 Ada 35
6.11 Modula 35
6.12 Modula 2 36
6.13 Other Language Developments 36
6.13.1 Logo 36
6.13.2 Postscript, TeX and LaTeX 37
6.13.3 Prolog 37
6.13.4 SQL 37
6.13.5 ICON 37
6.14 Object Orientated Programming – OOP 38
6.14.1 Oberon and Oberon 2 38
6.14.2 Smalltalk 39

6.14.3 C++ 39
6.15 Fortran 90 40
6.16 The Future and Further Sources 40
6.16.1 Fortran 1996 40
6.16.2 High Performance Fortran – HPF 41
6.16.3 Network Sources 41
6.17 Summary 42
6.18 Bibliography 42
7 Introduction to Programming 47
7.1 Elements of a programming language 49
7.2 Variables — name, type and value 52
7.3 Notes 54
7.4 Some more Fortran rules 55
7.5 Good Programming Guidelines 56
7.6 Fortran Character set 56
7.6.1 Notes 56
7.7 Problems 56
8 Introduction to Arithmetic 58
8.1 Rounding and truncation 61
8.2 Example 1: Time taken for light to travel from the Sun to Earth 63
8.3 The PARAMETER statement. 64
8.4 Precision and size of numbers 65
8.5 Health Warning: Optional Reading, Beginners are Advised to Leave until Later 67
8.5.1 Selecting different INTEGER Kinds 69
8.5.2 Selecting different REAL Kinds 69
8.5.3 Specifying Kind Types for Literal Integer and Real Constants 70
8.5.4 Positional Number Systems 70
8.5.5 Bit Data Type and Representation Model. 71
8.5.6 Integer Data Type and Representation Model 71
8.5.7 Real Data Type and Representation Model 72
8.5.8 IEEE 754 72
8.5.9 Example 2: Testing the numerical representation of different kind types on a system 72
8.5.10 Example 3: Binary Representation of Different Integer Kind Type Numbers . . 75
8.5.11 Summary of how to select the appropriate KIND type 77
8.6 Summary 77
8.7 Problems 78
8.8 Bibliography 80
9 Arrays 1: Some Fundamentals 81
9.1 Tables of data 82

9.1.1 Telephone directory . . . 82
9.1.2 Book catalogue . . . 82
9.1.3 Examination marks or results . . . 83
9.1.4 Monthly rainfall . . . 83
9.2 Arrays in Fortran . . . 84
9.3 The DIMENSION Attribute . . . 84
9.4 An index . . . 84
9.5 Control structure . . . 85
9.6 Monthly Rainfall . . . 85
9.6.1 Example 1: Rainfall . . . 86
9.7 People's Weights . . . 87
9.7.1 Example 2: People's Weights . . . 88
9.8 Summary . . . 88
9.9 Problems . . . 89

10 Arrays 2: Further Examples . . . 90

10.1 Higher dimension arrays . . . 91
10.1.1 Example 1: A Map . . . 91
10.1.2 Example 2: Booking arrangements in a theatre or cinema 92
10.2 Additional forms of the DIMENSION attribute and DO loop statement . 93
10.2.1 Example 3: Voltage from -20 to +20 volts . . . 93
10.2.2 Example 4: Longitude from -180 to +180 . . . 94
10.2.3 Notes . . . 94
10.3 The DO loop and straight repetition . . . 94
10.3.1 Example 5: Table of Temperatures . . . 94
10.3.2 Example 6: Means and Standard Deviations . . . 95
10.4 Summary . . . 96
10.5 Problems . . . 97

11 Arrays 3: Further Examples . . . 98

11.1 Terminology . . . 99
11.1.1 Rank . . . 99
11.1.2 Bounds . . . 99
11.1.3 Extent . . . 99
11.1.4 Size . . . 99
11.1.5 Shape . . . 99
11.1.6 Conformable . . . 99
11.2 Whole array manipulation . . . 99
11.2.1 Assignment . . . 100
11.2.2 Expressions . . . 100
11.3 Array Sections . . . 102
11.3.1 Example 1: Ages . . . 102

11.3.2 Example 2: Examination Results 102
11.4 Allocatable Arrays 102
11.4.1 Example 3: Height Above Sea Level 103
11.5 Array Element Ordering 104
11.5.1 Array Element Ordering and Physical and Virtual Memory 105
11.6 Array Constructors 105
11.7 Masked Array Assignment and the WHERE Statement. 106
11.7.1 Notes 107
11.8 Summary 107
11.9 Problems 107

12 Output 108

12.1 Integers, I format 109
12.2 Reals, F format 110
12.3 Reals, E format 111
12.4 Spaces 112
12.5 Alphanumeric or character format, A 113
12.6 Common mistakes 113
12.7 OPEN (and CLOSE) 114
12.7.1 The OPEN statement 114
12.7.2 Writing 115
12.8 Repetition 116
12.9 Some more examples 118
12.10 Implied DO loops 119
12.11 Formatting for a line-printer 120
12.11.1 Mechanics of carriage control 121
12.11.2 Generating a new line, on both line-printers and terminals. 122
12.12 Summary 123
12.13 Problems 123

13 Reading in Data 125

13.1 Fixed fields on input 126
13.1.1 Integers, the I format 126
13.1.2 Reals, the F and E formats 126
13.2 Blanks, nulls and zeros 129
13.3 Characters 129
13.4 Skipping spaces and lines 130
13.5 Reading 130
13.6 File manipulation again 131
13.7 Errors when reading 132
13.8 Summary 132

13.9 Problems 132

14 Functions 134

14.1 An Introduction to Predefined Functions and Their Use 135

14.1.1 Example 1: Simple function usage 136

14.2 Generic Functions 136

14.2.1 Example 2: The ABS Generic function 136

14.3 Elemental Functions 137

14.3.1 Example 3: Elemental Function Use 137

14.4 Transformational Functions 137

14.4.1 Example 4: Simple Transformational Use 137

14.4.2 Example 5: Intrinsic DOT_PRODUCT use 138

14.5 Notes on Function Usage 138

14.6 Example 6: Easter 138

14.7 Complete List of Predefined Functions 140

14.7.1 Inquiry Functions 140

14.7.2 Transfer and Conversion Functions 141

14.7.3 Computational Functions 141

14.7.4 Array Functions 141

14.7.5 Pre-Defined Subroutines 142

14.8 Supplying your own functions 142

14.8.1 Example 7: Simple User Defined Function 142

14.9 An Introduction to the Scope of Variables and Local Variables 144

14.10 Recursive Functions 145

14.10.1 Example 8: Recursive Factorial Evaluation 145

14.11 Example 9: Recursive version of GCD 146

14.12 Example 10: After Removing Recursion 147

14.13 Internal functions 148

14.13.1 Example 11: Stirling's Approximation 148

14.14 Resume 149

14.15 Function Syntax 150

14.16 Rules and Restrictions 150

14.17 Problems 150

14.18 Bibliography 151

14.18.1 Recursion and Problem Solving 151

15 Control Structures 152

15.1 Selection between courses of action 153

15.1.1 The BLOCK IF statement. 153

15.1.2 Example 1: Quadratic Roots 156

15.1.3 Note 157

15.1.4 Example 2: Date calculation 157

15.1.5 The CASE Statement . . . 158
15.1.6 Example 3: Simple calculator . . . 158
15.1.7 Example 4: Counting Vowels, Consonants, etc . . . 159
15.2 The three forms of the DO statement. . . . 160
15.2.1 Example 5: Sentinel Usage . . . 161
15.2.2 CYCLE and EXIT . . . 162
15.2.3 Example 6: e**x Evaluation . . . 162
15.2.4 Example 7: Wave Breaking on an Offshore Reef . . . 163
15.3 Summary . . . 165
15.4 Problems . . . 166
15.5 Bibliography . . . 169
16 Character . . . 170
16.1 Character Input . . . 172
16.2 Character Operators . . . 173
16.3 Character Sub-Strings . . . 174
16.4 Character functions . . . 176
16.5 Summary . . . 178
16.6 Problems . . . 178
17 Complex . . . 180
17.1 Example . . . 182
17.2 Complex and Kind Type . . . 183
17.3 Summary . . . 183
17.4 Problems . . . 183
18 Logical . . . 184
18.1 I/O . . . 187
18.2 Summary . . . 188
18.3 Problems . . . 188
19 User Defined Types . . . 189
19.1 Example 1 – Dates . . . 190
19.2 Type Definition . . . 190
19.3 Variable Definition . . . 191
19.4 Example 2 – Address lists . . . 191
19.5 Example 3: Nested User Defined Types . . . 192
19.6 Problems . . . 194
19.7 Bibliography . . . 194
20 Dynamic Data Structures . . . 195
20.1 Example 1: Simple Pointer Concepts . . . 196
20.2 Example 2: Singly linked list . . . 197

20.3 Other Dynamic Data Structures 199
20.4 Trees 200
20.4.1 Example 3: Perfectly Balanced Tree 200
20.5 Using Linked Lists for Sparse Matrix Problems 203
20.5.1 Inner Product of two Sparse Vectors 204
20.6 Data Structures Summary 208
20.7 Problems 208
20.8 Bibliography 209

21 Files 211

21.1 Files in Fortran 212
21.2 Summary of options on OPEN 214
21.3 More fool proof i/o 215
21.4 Summary 217
21.5 Problems 217

22 An Introduction to Subroutines 218

22.1 Simple Subroutine Example 219
22.2 Defining a subroutine 222
22.3 Referencing a subroutine 222
22.4 Dummy Arguments or Parameters, and Actual Arguments 222
22.5 Interface 222
22.6 Intent 223
22.7 Local Variables 224
22.7.1 Local Variables and the SAVE attribute 224
22.8 Scope of Variables 224
22.9 Status of the Action Carried out in the Subroutine 224
22.10 Why Bother? 225
22.11 Summary 225
22.12 Problems 225

23 Subroutines: 2 226

23.1 Example 1: Introduction to Arrays as Parameters 227
23.1.1 Explicit Shape Dummy Arrays 227
23.2 Example 2: Characters as parameters and assumed length dummy arguments 228
23.3 Example 3: Using Hoare's Quick Sort Algorithm 229
23.3.1 Note 2 – Intent Attribute 235
23.3.2 Note 3 – Explicit shape dummy array 235
23.3.3 Note 4 – Assumed Length Dummy Argument 235
23.3.4 Note 5 – Recursive Subroutine 235
23.3.5 Note 6 – Internal Subroutines and Scope 235
23.3.6 Note 7 – Flexible Design 236

23.4 Example 4: Rank two and higher arrays as parameters 236
23.4.1 Assumed Shape Arrays 236
23.5 Summary 238
23.6 Problems 238
23.7 Bibliography 239
24 An Introduction to Modules 241
24.1 Modules for global data 242
24.1.1 Example 1: Modules for Precision Specification and Constant Definition . 243
24.1.2 Note 243
24.1.3 Example 2: Constant Definition and Array Definition . . . 244
24.2 Modules for derived data types 245
24.2.1 Example 3: Person Data Type 245
24.3 Modules for explicit procedures interfaces 248
24.3.1 Example: Using QuickSort 248
24.4 Modules containing procedures 249
24.5 Example 4 – The Solution of Linear Equations Using Gaussian Elimination 251
24.5.1 Notes 256
24.6 Notes on Module Usage and Compilation 256
24.7 Summary 257
24.8 Problems 257
24.9 Bibliography 258
25 Formal Syntax and Some Additional Features 259
25.1 Program Units 260
25.2 Procedure – Function or Subroutine 260
25.2.1 Internal Procedure 260
25.3 Module 260
25.4 Executable Statements 261
25.5 Statement Ordering 261
25.6 Entities 262
25.7 Scope and Association 262
25.8 Modules and Scope 264
25.8.1 Public and Private Attributes 264
25.8.2 USE, ONLY and Rename 265
25.9 Keyword and Optional Arguments 266
25.10 Syntax Summary of Some Frequently used Fortran Constructs 267
25.10.1 Main Program 267
25.10.2 Subprogram 267
25.10.3 Module 268
25.10.4 Internal Procedure 268

25.10.5 Procedure heading . . . 268
25.10.6 Procedure ending . . . 268
25.10.7 Specification construct . . . 268
25.10.8 Derived Type definition . . . 268
25.10.9 Interface block . . . 269
25.10.10 Specification statement . . . 269
25.10.11 Type specification . . . 269
25.10.12 Attribute Specification . . . 269
25.10.13 Executable construct . . . 270
25.10.14 Action statement . . . 270
26 Case Studies . . . 271
26.1 Example 1 – Solving a System of First Order Ordinary Differential Equations using Runge-Kutta-Merson . . . 272
26.1.1 Note: Alternative form of the Allocate statement . . . 278
26.1.2 Note: Automatic arrays . . . 278
26.1.3 Note: Dummy Procedure Arguments . . . 279
26.2 Example 2 – Generic Procedures . . . 279
26.3 Example 3 – A Function that returns a variable length array . . . 285
26.4 Example 4 – Operator and Assignment Overloading . . . 286
26.5 Example 5: A Subroutine to Extract the Diagonal Elements of a Matrix . . . 287
26.6 Modules and Packaging . . . 288
26.7 Problems . . . 290
26.8 Bibliography . . . 290
27 Converting from Fortran 77 . . . 291
27.1 Deleted Features . . . 292
27.2 Obsolescent Features . . . 292
27.2.1 Arithmetic IF . . . 292
27.2.2 Real and Double precision DO Control Variables . . . 292
27.2.3 Shared DO termination and non ENDDO termination . . . 292
27.2.4 Alternate RETURN . . . 292
27.2.5 PAUSE Statement . . . 293
27.2.6 ASSIGN and assigned GOTO statements . . . 293
27.2.7 Assigned FORMAT statements . . . 293
27.2.8 H Editing . . . 293
27.3 Better Alternatives . . . 293
28 Miscellanea . . . 295
28.1 Program Development and Software Engineering . . . 296
28.1.1 Modules . . . 296
28.1.2 Programming Style – Programs should be Easy to Read . . . 297
28.1.3 Programming Style – Programs should Behave Well . . . 297

28.2 Data Structures 298
28.3 Algorithms 298
28.4 Recursion 298
28.5 Structured Programming and the GOTO Statement 298
28.6 Efficiency, space time trade off 299
28.7 Program Testing 299
28.8 Simple Debugging Techniques 300
28.9 Software Tools 300
28.9.1 Cross Referencing 300
28.9.2 Pretty print 300
28.9.3 NAGWare f90 Tools 300
28.10 Numerical Software Sources 300
28.11 Coda 301
28.12 Bibliography: All sources (bar one) taken from comp.software-eng. 302
28.12.1 Software Engineering 302
28.12.2 Programming Style 302
28.12.3 Software Testing 302
28.12.4 Fun 302
Glossary 303
Appendix A, Sample Program Examples 309
Appendix B, ASCII Character Set 312
Appendix C, Intrinsic Functions and Procedures 313
Appendix D, English and Latin Texts 340
Appendix E, Coded Text Extracts 341
Appendix F, NAG 342
Appendix G, Annex D, ISO/IEC 1539: 1991 (E) 344
Index 366