TABLE OF CONTENTS

Preface xix

1 Introduction to Computers and The Fortran Language 1

1.1 The Computer 2
1.1.1. The CPU / 1.1.2. Memory / 1.1.3. Input and Output Devices

1.2 Data Representation in a Computer 4
1.2.1. The Binary Number System / 1.2.2. Octal and Hexadecimal Representations of Binary Numbers / 1.2.3. Types of Data Stored in Memory

1.3 Computer Languages 12

1.4 The History of the Fortran Language 13

1.5 The Evolution of Fortran 16

1.6 Summary 19
1.6.1. Exercises

2 Basic Elements of Fortran 22

2.1 Introduction 22

2.2 The Fortran Character Set 23

2.3 The Structure of a Fortran Statement 23

2.4 The Structure of a Fortran Program 24
2.4.1. The Declaration Section / 2.4.2. The Execution Section / 2.4.3. The Termination Section / 2.4.4. Program Style / 2.4.5. Compiling, Linking, and Executing the Fortran Program

2.5 Constants and Variables 28
2.5.1. Integer Constants and Variables / 2.5.2. Real Constants and Variables / 2.5.3. Character Constants and Variables / 2.5.4. Default and Explicit Variable Typing / 2.5.5. Keeping Constants Consistent in a Program

2.6 Assignment Statements and Arithmetic Calculations 36
2.6.1. Integer Arithmetic / 2.6.2. Real Arithmetic / 2.6.3. Hierarchy of Operations / 2.6.4. Mixed-Mode Arithmetic / 2.6.5. Mixed-Mode Arithmetic and Exponentiation

2.7 Intrinsic Functions 47

2.8 List-Directed Input and Output Statements 49

2.9 Initialization of Variables 55

2.10 The `IMPLICIT NONE` Statement 57

2.11 Program Examples 58

2.12 Debugging Fortran Programs 66

2.13 Summary 68
2.13.1. Summary of Good Programming Practice / 2.13.2. Summary of Fortran Statements / 2.13.3. Exercises

3 Program Design and Branching Structures 81

3.1 Introduction to Top-Down Design Techniques 82

3.2 Use of Pseudocode and Flowcharts 86

3.3 Logical Constants, Variables, and Operators 89
3.3.1. Logical Constants and Variables / 3.3.2. Assignment Statements and Logical Calculations / 3.3.3. Relational Operators / 3.3.4. Combinational Logic Operators / 3.3.5. Logical Values in Input and Output Statements / 3.3.6. The Significance of Logical Variables and Expressions

3.4 Control Constructs: Branches 94
3.4.1. The Block `IF` Construct / 3.4.2. The `ELSE` and `ELSE IF` Clauses / 3.4.3. Examples Using Block `IF` Constructs / 3.4.4. Named Block `IF` Constructs / 3.4.5. Notes Concerning the Use of Block `IF` Constructs / 3.4.6. The Logical `IF` Statement / 3.4.7. The `SELECT CASE` Construct

3.5 More on Debugging Fortran Programs 118

3.6 Summary 119
3.6.1. Summary of Good Programming Practice / 3.6.2. Summary of Fortran Statements and Constructs / 3.6.3. Exercises

4 Loops and Character Manipulation 126

4.1 Control Constructs: Loops 126
4.1.1 The While Loop / 4.1.2 The `DO WHILE` Loop / 4.1.3 The Iterative or Counting Loop / 4.1.4 The `CYCLE` and `EXIT` Statements / 4.1.5 Named Loops / 4.1.6 Nesting Loops and Block `IF` Constructs

4.2 Character Assignments and Character Manipulations 154
4.2.1 Character Assignments / 4.2.2 Substring Specifications / 4.2.3 The Concatenation (`//`) Operator / 4.2.4 Relational Operators with Character Data / 4.2.5 Character Intrinsic Functions

4.3 Debugging Fortran Loops 168

4.4 Summary 169
4.4.1 Summary of Good Programming Practice / 4.4.2 Summary of Fortran Statements and Constructs / 4.4.3 Exercises

5 Basic I/O Concepts 180

5.1 Formats and Formatted WRITE Statements 180

5.2 Output Devices 182
5.2.1 Control Characters in Printer Output

5.3 Format Descriptors 184
5.3.1 Integer Output—The I Descriptor / 5.3.2 Real Output—The F Descriptor / 5.3.3 Real Output—The E Descriptor / 5.3.4 True Scientific Notation—The ES Descriptor / 5.3.5 Logical Output—The L Descriptor / 5.3.6 Character Output—The A Descriptor / 5.3.7 Horizontal Positioning—The X and T Descriptor / 5.3.8 Repeating Groups of Format Descriptors / 5.3.9 Changing Output Lines—The Slash (/) Descriptor / 5.3.10 How Formats are Used During WRITEs

5.4 Formatted READ Statements 205
5.4.1 Integer Input—The I Descriptor / 5.4.2 Real Input—The F Descriptor / 5.4.3 Logical Input—The L Descriptor / 5.4.4 Character Input—The A Descriptor / 5.4.5 Horizontal Positioning—The X and T Descriptors / 5.4.6 Vertical Positioning—The Slash (/) Descriptor / 5.4.7 How Formats are Used During READs

5.5 An Introduction to Files and File Processing 211
5.5.1 The OPEN Statement / 5.5.2 The CLOSE Statement / 5.5.3 READs and WRITEs to Disk Files / 5.5.4 The IOSTAT= and IOMSG= Clauses in the READ Statement / 5.5.5 File Positioning

5.6 Summary 232
5.6.1 Summary of Good Programming Practice / 5.6.2 Summary of Fortran Statements and Structures / 5.6.3 Exercises

6 Introduction to Arrays 245

6.1 Declaring Arrays 246

6.2 Using Array Elements in Fortran Statements 247
6.2.1 Array Elements are Just Ordinary Variables / 6.2.2 Initialization of Array Elements / 6.2.3 Changing the Subscript Range of an Array / 6.2.4 Out-of-Bounds Array Subscripts / 6.2.5 The Use of Named Constants with Array Declarations

6.3 Using Whole Arrays and Array Subsets in Fortran Statements 261
6.3.1 Whole Array Operations / 6.3.2 Array Subsets

6.4 Input and Output 265
6.4.1 Input and Output of Array Elements / 6.4.2 The Implied DO Loop / 6.4.3 Input and Output of Whole Arrays and Array Sections

6.5 Example Problems 271

6.6 When Should You Use an Array? 287

6.7 Summary 289
6.7.1 Summary of Good Programming Practice / 6.7.2 Summary of Fortran Statements and Constructs / 6.7.3 Exercises

7 Introduction to Procedures 297

7.1 Subroutines 299
7.1.1 Example Problem—Sorting / 7.1.2 The INTENT Attribute / 7.1.3 Variable Passing in Fortran: The Pass-By-Reference Scheme / 7.1.4 Passing Arrays to Subroutines / 7.1.5 Passing Character Variables to Subroutines / 7.1.6 Error Handling in Subroutines / 7.1.7 Examples

7.2 Sharing Data Using Modules 320

7.3 Module Procedures 328
7.3.1 Using Modules to Create Explicit Interfaces

7.4 Fortran Functions 331
7.4.1 Unintended Side Effects in Functions / 7.4.2 Using Functions with Deliberate Side Effects

7.5 Passing Procedures as Arguments to Other Procedures 339
7.5.1 Passing User-Defined Functions as Arguments / 7.5.2 Passing Subroutines as Arguments

7.6 Summary 344
7.6.1 Summary of Good Programming Practice / 7.6.2 Summary of Fortran Statements and Structures / 7.6.3 Exercises

8 Additional Features of Arrays 360

8.1 2D or Rank 2 Arrays 360
8.1.1 Declaring Rank 2 Arrays / 8.1.2 Rank 2 Array Storage / 8.1.3 Initializing Rank 2 Arrays / 8.1.4 Example Problem / 8.1.5 Whole Array Operations and Array Subsets

8.2 Multidimensional or Rank *n* Arrays 372

8.3 Using Fortran Intrinsic Functions with Arrays 375
8.3.1 Elemental Intrinsic Functions / 8.3.2 Inquiry Intrinsic Functions / 8.3.3 Transformational Intrinsic Functions

8.4 Masked Array Assignment: The WHERE Construct 378
8.4.1 The WHERE Construct / 8.4.2 The WHERE Statement

8.5 The FORALL Construct 381
8.5.1 The Form of the FORALL Construct / 8.5.2 The Significance of the FORALL Construct / 8.5.3 The FORALL Statement

8.6 Allocatable Arrays 383
8.6.1 Fortran Allocatable Arrays / 8.6.2 Using Fortran Allocatable Arrays in Assignment Statements

8.7 Summary 393
8.7.1 Summary of Good Programming Practice / 8.7.2 Summary of Fortran Statements and Constructs / 8.7.3 Exercises

9 Additional Features of Procedures 404

9.1 Passing Multidimensional Arrays to Subroutines and Functions 404
9.1.1 Explicit Shape Dummy Arrays / 9.1.2 Assumed-Shape Dummy Arrays / 9.1.3 Assumed-Size Dummy Arrays

9.2 The `SAVE` Attribute and Statement 417

9.3 Allocatable Arrays in Procedures 421

9.4 Automatic Arrays in Procedures 422
9.4.1 Comparing Automatic Arrays and Allocatable Arrays / 9.4.2 Example Program

9.5 Allocatable Arrays as Dummy Arguments in Procedures 430
9.5.1 Allocatable Dummy Arguments / 9.5.2 Allocatable Functions

9.6 Pure and Elemental Procedures 434
9.6.1 Pure Procedures / 9.6.2 Elemental Procedures / 9.6.3 Impure Elemental Procedures

9.7 Internal Procedures 436

9.8 Submodules 438

9.9 Summary 446
9.9.1 Summary of Good Programming Practice / 9.9.2 Summary of Fortran Statements and Structures / 9.9.3 Exercises

10 More about Character Variables 457

10.1 Character Comparison Operations 458
10.1.1 The Relational Operators with Character Data / 10.1.2 The Lexical Functions `LLT`, `LLE`, `LGT`, and `LGE`

10.2 Intrinsic Character Functions 463

10.3 Passing Character Variables to Subroutines and Functions 465

10.4 Variable-Length Character Functions 471

10.5 Internal Files 473

10.6 Example Problems 474

10.7 Summary 479
10.7.1 Summary of Good Programming Practice / 10.7.2 Summary of Fortran Statements and Structures / 10.7.3 Exercises

11 Additional Intrinsic Data Types 485

11.1 Alternate Kinds of the REAL Data Type 485
11.1.1 Kinds of REAL Constants and Variables / 11.1.2 Determining the KIND of a Variable / 11.1.3 Selecting Precision in a Processor-Independent Manner / 11.1.4 Determining the KINDs of Data Types on a Particular Processor / 11.1.5 Mixed-Mode Arithmetic / 11.1.6 Higher Precision Intrinsic Functions / 11.1.7 When to Use High-Precision Real Values / 11.1.8 Solving Large Systems of Simultaneous Linear Equations

11.2 Alternate Lengths of the INTEGER Data Type 509

11.3 Alternate Kinds of the CHARACTER Data Type 511

11.4 The COMPLEX Data Type 512
11.4.1 Complex Constants and Variables / 11.4.2 Initializing Complex Variables / 11.4.3 Mixed-Mode Arithmetic / 11.4.4 Using Complex Numbers with Relational Operators / 11.4.5 COMPLEX Intrinsic Functions

11.5 Summary 522
11.5.1 Summary of Good Programming Practice / 11.5.2 Summary of Fortran Statements and Structures / 11.5.3 Exercises

12 Derived Data Types 527

12.1 Introduction to Derived Data Types 527

12.2 Working with Derived Data Types 529

12.3 Input and Output of Derived Data Types 529

12.4 Declaring Derived Data Types in Modules 531

12.5 Returning Derived Types from Functions 540

12.6 Dynamic Allocation of Derived Data Types 544

12.7 Parameterized Derived Data Types 545

12.8 Type Extension 546

12.9 Type-Bound Procedures 548

12.10 The ASSOCIATE Construct 552

12.11 Summary 553
12.11.1 Summary of Good Programming Practice / 12.11.2 Summary of Fortran Statements and Structures / 12.11.3 Exercises

13 Advanced Features of Procedures and Modules 561

13.1 Scope and Scoping Units 562

13.2 Blocks 567

13.3 Recursive Procedures 568

13.4 Keyword Arguments and Optional Arguments 571

13.5 Procedure Interfaces and Interface Blocks 577
13.5.1 Creating Interface Blocks / 13.5.2 Notes on the Use of Interface Blocks

13.6 Generic Procedures 581
13.6.1 User-Defined Generic Procedures / 13.6.2 Generic Interfaces for Procedures in Modules / 13.6.3 Generic Bound Procedures

13.7 Extending Fortran with User-Defined Operators and Assignments 594

13.8 Bound Assignments and Operators 607

13.9 Restricting Access to the Contents of a Module 607

13.10 Advanced Options of the `USE` Statement 611

13.11 Intrinsic Modules 615

13.12 Access to Command Line Arguments and Environment Variables 615
13.12.1 Access to Command Line Arguments / 13.12.2 Retrieving Environment Variables

13.13 The `VOLATILE` Attribute and Statement 618

13.14 Summary 619
13.14.1 Summary of Good Programming Practice / 13.14.2 Summary of Fortran Statements and Structures / 13.14.3 Exercises

14 Advanced I/O Concepts 633

14.1 Additional Format Descriptors 633
14.1.1 Additional Forms of the `E` and `ES` Format Descriptors / 14.1.2 Engineering Notation—The `EN` Descriptor / 14.1.3 Double-Precision Data—The `D` Descriptor / 14.1.4 The Generalized (`G`) Format Descriptor / 14.1.5 The `G0` Format Descriptor / 14.1.6 The Binary, Octal, and Hexadecimal (`B`, `O`, and `Z`) Descriptors / 14.1.7 The TAB Descriptors / 14.1.8 The Colon (`:`) Descriptor / 14.1.9 Scale Factors—The `P` Descriptor / 14.1.10 The `SIGN` Descriptors / 14.1.11 Blank Interpretation: The `BN` and `BZ` Descriptors / 14.1.12 Rounding Control: The `RU`, `RD`, `RZ`, `RN`, `RC`, and `RP` Descriptors / 14.1.13 Decimal Specifier: The `DC` and `DP` Descriptors

14.2 Defaulting Values in List-Directed Input 642

14.3 Detailed Description of Fortran I/O Statements 644
14.3.1 The `OPEN` Statement / 14.3.2 The `CLOSE` Statement / 14.3.3 The `INQUIRE` Statement / 14.3.4 The `READ` Statement / 14.3.5 Alternate Form of the `READ` Statement / 14.3.6 The `WRITE` Statement / 14.3.7 The `PRINT` Statement / 14.3.8 File Positioning Statements / 14.3.9 The `ENDFILE` Statement / 14.3.10 The `WAIT` Statement / 14.3.11 The `FLUSH` Statement

14.4 Namelist I/O 668

14.5 Unformatted Files 671

14.6 Direct Access Files 673

14.7 Stream Access Mode 678

14.8 Nondefault I/O for Derived Types 678

14.9 Asynchronous I/O 687
14.9.1. Performing Asynchronous I/O / 14.9.2. Problems with Asynchronous I/O

14.10 Access to Processor-Specific I/O System Information 689

14.11 Summary 690
14.11.1 Summary of Good Programming Practice / 14.11.2 Summary of Fortran Statements and Structures / 14.11.3 Exercises

15 Pointers and Dynamic Data Structures 698

15.1 Pointers and Targets 699
15.1.1 Pointer Assignment Statements / 15.1.2 Pointer Association Status

15.2 Using Pointers in Assignment Statements 705

15.3 Using Pointers with Arrays 707

15.4 Dynamic Memory Allocation with Pointers 709

15.5 Using Pointers as Components of Derived Data Types 712

15.6 Arrays of Pointers 725

15.7 Using Pointers in Procedures 727
15.7.1 Using the INTENT Attribute with Pointers / 15.7.2 Pointer-valued Functions

15.8 Procedure Pointers 733

15.9 Binary Tree Structures 736
15.9.1 The Significance of Binary Tree Structures / 15.9.2 Building a Binary Tree Structure

15.10 Summary 756
15.10.1 Summary of Good Programming Practice / 15.10.2 Summary of Fortran Statements and Structures / 15.10.3 Exercises

16 Object-Oriented Programming in Fortran 763

16.1 An Introduction to Object-Oriented Programming 764
16.1.1 Objects / 16.1.2 Messages / 16.1.3 Classes / 16.1.4 Class Hierarchy and Inheritance / 16.1.5 Object-Oriented Programming

16.2 The Structure of a Fortran Class 769

16.3 The `CLASS` Keyword 770

16.4 Implementing Classes and Objects in Fortran 772
16.4.1 Declaring Fields (Instance Variables) / 16.4.2 Creating Methods / 16.4.3 Creating (Instantiating) Objects from a Class

16.5 First Example: A `timer` Class 775
16.5.1 Implementing the `timer` Class / 16.5.2 Using the `timer` Class / 16.5.3 Comments on the `timer` Class

16.6 Categories of Methods 780

16.7 Controlling Access to Class Members 789

16.8 Finalizers 790

16.9 Inheritance and Polymorphism 794
16.9.1 Superclasses and Subclasses / 16.9.2 Defining and Using Subclasses / 16.9.3 The Relationship between Superclass Objects and Subclass Objects / 16.9.4 Polymorphism / 16.9.5 The `SELECT TYPE` Construct

16.10 Preventing Methods from Being Overridden in Subclasses 809

16.11 Abstract Classes 809

16.12 Summary 831
16.12.1 Summary of Good Programming Practice / 16.12.2 Summary of Fortran Statements and Structures / 16.12.3 Exercises

17 Coarrays and Parallel Processing 837

17.1 Parallel Processing in Coarray Fortran 838

17.2 Creating a Simple Parallel Program 839

17.3 Coarrays 841

17.4 Synchronization between Images 843

17.5 Example: Sorting a Large Data Set 850

17.6 Allocatable Coarrays and Derived Data Types 856

17.7 Passing Coarrays to Procedures 857

17.8 Critical Sections 858

17.9 The Perils of parallel Programming 859

17.10 Summary 863
17.10.1 Summary of Good Programming Practice / 17.10.2 Summary of Fortran Statements and Structures / 17.10.3 Exercises

18 Redundant, Obsolescent, and Deleted Fortran Features 869

18.1 Pre-Fortran 90 Character Restrictions 870

18.2 Obsolescent Source Form 870

18.3 Redundant Data Type 871

18.4 Older, Obsolescent, and/or Undesirable Specification Statements 872
18.4.1 Pre-Fortran 90 Specification Statements / 18.4.2 The `IMPLICIT` Statement / 18.4.3 The `DIMENSION` Statement / 18.4.4 The `DATA` Statement / 18.4.5 The `PARAMETER` Statement

18.5 Sharing Memory Locations: COMMON and EQUIVALENCE 875
18.5.1 COMMON Blocks / 18.5.2 Initializing Data in COMMON Blocks: The BLOCK DATA Subprogram / 18.5.3 The Unlabeled COMMON Statement / 18.5.4 The EQUIVALENCE Statement

18.6 Undesirable Subprogram Features 882
18.6.1 Alternate Subroutine Returns / 18.6.2 Alternate Entry Points / 18.6.3 The Statement Function / 18.6.4 Passing Intrinsic Functions as Arguments

18.7 Miscellaneous Execution Control Features 889
18.7.1 The PAUSE Statement / 18.7.2 Arguments Associated with the STOP Statement / 18.7.3 The END Statement

18.8 Obsolete Branching and Looping Structures 892
18.8.1 The Arithmetic IF Statement / 18.8.2 The Unconditional GO TO Statement / 18.8.3 The Computed GO TO Statement / 18.8.4 The Assigned GO TO Statement / 18.8.5 Older Forms of DO Loops

18.9 Redundant Features of I/O Statements 896

18.10 Summary 897
18.10.1 Summary of Good Programming Practice / 18.10.2 Summary of Fortran Statements and Structures

Appendixes

A. The ASCII Character Set 903

B. Fortran/C Interoperability 904
B.1. Declaring Interoperable Data Types / B.2. Declaring Interoperable Procedures / B.3. Sample Programs—Fortran Calling C / B.4. Sample Programs—C Calling Fortran

C. Fortran Intrinsic Procedures 914
C.1. Classes of Intrinsic Procedures / C.2. Alphabetical List of Intrinsic Procedures / C.3. Mathematical and Type Conversion Intrinsic Procedures / C.4. Kind and Numeric Processor Intrinsic Functions / C.5. System Environment Procedures / C.6. Bit Intrinsic Procedures / C.7. Character Intrinsic Functions / C.8. Array and Pointer Intrinsic Functions / C.9. Miscellaneous Inquiry Functions / C.10. Miscellaneous Procedures / C.11. Coarray Functions

D. Order of Statements in a Fortran Program 961

E. Glossary 963

F. Answers to Quizzes 984

Index 1002

Summary of Selected Fortran Statements and Structures 1022