

T A B L E O F C O N T E N T S

Preface	xxi
1 Introduction to Computers and the Fortran Language	1
1.1 The Computer	2
<i>1.1.1 The CPU / 1.1.2 Main and Secondary Memory /</i>	
<i>1.1.3 Input and Output Devices</i>	
1.2 Data Representation in a Computer	4
<i>1.2.1 The Binary Number System / 1.2.2 Octal and Hexadecimal</i>	
<i>Representations of Binary Numbers / 1.2.3 Types of Data Stored</i>	
<i>in Memory</i>	
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
<i>1.6.1 Exercises</i>	
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
<i>2.4.1 The Declaration Section / 2.4.2 The Execution Section /</i>	
<i>2.4.3 The Termination Section / 2.4.4 Program Style /</i>	
<i>2.4.5 Compiling, Linking, and Executing the Fortran Program</i>	
2.5 Constants and Variables	28
<i>2.5.1 Integer Constants and Variables / 2.5.2 Real</i>	
<i>Constants and Variables / 2.5.3 Character Constants and</i>	
<i>Variables / 2.5.4 Default and Explicit Variable Typing /</i>	
<i>2.5.5 Keeping Constants Consistent in a Program</i>	

2.6	Assignment Statements and Arithmetic Calculations	36
	<i>2.6.1 Integer Arithmetic / 2.6.2 Real Arithmetic /</i>	
	<i>2.6.3 Hierarchy of Operations / 2.6.4 Mixed-Mode</i>	
	<i>Arithmetic / 2.6.5 Mixed-Mode Arithmetic and Exponentiation</i>	
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
	<i>2.13.1 Summary of Good Programming Practice /</i>	
	<i>2.13.2 Summary of Fortran Statements / 2.13.3 Exercises</i>	
3	Program Design and Branching Structures	82
3.1	Introduction to Top-Down Design Techniques	83
3.2	Use of Pseudocode and Flowcharts	87
3.3	Logical Constants, Variables, and Operators	90
	<i>3.3.1 Logical Constants and Variables / 3.3.2 Assignment</i>	
	<i>Statements and Logical Calculations / 3.3.3 Relational</i>	
	<i>Operators / 3.3.4 Combinational Logic Operators /</i>	
	<i>3.3.5 Logical Values in Input and Output Statements /</i>	
	<i>3.3.6 The Significance of Logical Variables and Expressions</i>	
3.4	Control Constructs: Branches	96
	<i>3.4.1 The Block IF Construct / 3.4.2 The ELSE and ELSE IF</i>	
	<i>Clauses / 3.4.3 Examples Using Block IF Constructs /</i>	
	<i>3.4.4 Named Block IF Constructs / 3.4.5 Notes Concerning the</i>	
	<i>Use of Block IF Constructs / 3.4.6 The Logical IF Statement /</i>	
	<i>3.4.7 The SELECT CASE Construct</i>	
3.5	More on Debugging Fortran Programs	119
3.6	Summary	120
	<i>3.6.1 Summary of Good Programming Practice / 3.6.2 Summary</i>	
	<i>of Fortran Statements and Constructs / 3.6.3 Exercises</i>	
4	Loops and Character Manipulation	128
4.1	Control Constructs: Loops	128
	<i>4.1.1 The While Loop / 4.1.2 The DO WHILE Loop /</i>	
	<i>4.1.3 The Iterative or Counting Loop / 4.1.4 The CYCLE and</i>	
	<i>EXIT Statements / 4.1.5 Named Loops / 4.1.6 Nesting Loops</i>	
	<i>and Block IF Constructs</i>	

4.2	Character Assignments and Character Manipulations	156
	<i>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</i>	
4.3	Debugging Fortran Loops	170
4.4	Summary	171
	<i>4.4.1 Summary of Good Programming Practice / 4.4.2 Summary of Fortran Statements and Constructs / 4.4.3 Exercises</i>	
5	Basic I/O Concepts	183
5.1	Formats and Formatted WRITE Statements	183
5.2	Output Devices	185
5.3	Format Descriptors	188
	<i>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 Descriptors / 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</i>	
5.4	Formatted READ Statements	209
	<i>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</i>	
5.5	An Introduction to Files and File Processing	215
	<i>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</i>	
5.6	Summary	237
	<i>5.6.1 Summary of Good Programming Practice / 5.6.2 Summary of Fortran Statements and Structures / 5.6.3 Exercises</i>	
6	Introduction to Arrays	251
6.1	Declaring Arrays	252
6.2	Using Array Elements in Fortran Statements	253
	<i>6.2.1 Array Elements Are Just Ordinary Variables / 6.2.2 Initialization of Array Elements / 6.2.3 Changing the</i>	

	<i>Subscript Range of an Array / 6.2.4 Out-of-Bounds Array Subscripts / 6.2.5 The Use of Named Constants with Array Declarations</i>	
6.3	Using Whole Arrays and Array Subsets in Fortran Statements	267
	<i>6.3.1 Whole Array Operations / 6.3.2 Array Subsets</i>	
6.4	Input and Output	271
	<i>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</i>	
6.5	Example Problems	277
6.6	When Should You Use an Array?	294
6.7	Summary	295
	<i>6.7.1 Summary of Good Programming Practice / 6.7.2 Summary of Fortran Statements and Constructs / 6.7.3 Exercises</i>	
7	Introduction to Procedures	305
7.1	Subroutines	307
	<i>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</i>	
7.2	Sharing Data Using Modules	329
7.3	Module Procedures	337
	<i>7.3.1 Using Modules to Create Explicit Interfaces</i>	
7.4	Fortran Functions	340
	<i>7.4.1 Unintended Side Effects in Functions</i>	
7.5	Passing Procedures as Arguments to Other Procedures	348
	<i>7.5.1 Passing User-Defined Functions as Arguments / 7.5.2 Passing Subroutines as Arguments</i>	
7.6	Summary	353
	<i>7.6.1 Summary of Good Programming Practice / 7.6.2 Summary of Fortran Statements and Structures / 7.6.3 Exercises</i>	
8	Additional Features of Arrays	370
8.1	Two-Dimensional or Rank-2 Arrays	370
	<i>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</i>	
8.2	Multidimensional or Rank-<i>n</i> Arrays	382

8.3	Using Fortran Intrinsic Functions with Arrays	385
	<i>8.3.1 Elemental Intrinsic Functions / 8.3.2 Inquiry Intrinsic Functions / 8.3.3 Transformational Intrinsic Functions</i>	
8.4	Masked Array Assignment: The WHERE Construct	389
	<i>8.4.1 The WHERE Construct / 8.4.2 The WHERE Statement</i>	
8.5	The FORALL Construct	391
	<i>8.5.1 The Form of the FORALL Construct / 8.5.2 The Significance of the FORALL Construct / 8.5.3 The FORALL Statement</i>	
8.6	Allocatable Arrays	394
	<i>8.6.1 Fortran 95 Allocatable Arrays / 8.6.2 Fortran 2003 Allocatable Arrays / 8.6.3 Using Fortran 2003 Allocatable Arrays in Assignment Statements</i>	
8.7	Summary	403
	<i>8.7.1 Summary of Good Programming Practice / 8.7.2 Summary of Fortran Statements and Constructs / 8.7.3 Exercises</i>	
9	Additional Features of Procedures	414
9.1	Passing Multidimensional Arrays to Subroutines and Functions	414
	<i>9.1.1 Explicit-Shape Dummy Arrays / 9.1.2 Assumed-Shape Dummy Arrays / 9.1.3 Assumed-Size Dummy Arrays</i>	
9.2	The SAVE Attribute and Statement	427
9.3	Allocatable Arrays in Procedures	432
9.4	Automatic Arrays in Procedures	432
	<i>9.4.1 Comparing Automatic Arrays and Allocatable Arrays / 9.4.2 Example Program</i>	
9.5	Allocatable Arrays in Fortran 2003 Procedures	440
	<i>9.5.1 Allocatable Dummy Arguments / 9.5.2 Allocatable Functions</i>	
9.6	Pure and Elemental Procedures	444
	<i>9.6.1 Pure Procedures / 9.6.2 Elemental Procedures</i>	
9.7	Internal Procedures	446
9.8	Summary	448
	<i>9.8.1 Summary of Good Programming Practice / 9.8.2 Summary of Fortran Statements and Structures / 9.8.3 Exercises</i>	
10	More about Character Variables	457
10.1	Character Comparison Operations	458
	<i>10.1.1 The Relational Operators with Character Data / 10.1.2 The Lexical Functions LLT, LLE, LGT, and LGE</i>	
10.2	Intrinsic Character Functions	463
10.3	Passing Character Variables to Subroutines and Functions	466

10.4	Variable-Length Character Functions	471
10.5	Internal Files	474
10.6	Example Problem	475
10.7	Summary	480
	<i>10.7.1 Summary of Good Programming Practice /</i>	
	<i>10.7.2 Summary of Fortran Statements and Structures /</i>	
	<i>10.7.3 Exercises</i>	
11	Additional Intrinsic Data Types	487
11.1	Alternative Kinds of the REAL Data Type	487
	<i>11.1.1 Kinds of REAL Constants and Variables /</i>	
	<i>11.1.2 Determining the KIND of a Variable / 11.1.3 Selecting</i>	
	<i>Precision in a Processor-Independent Manner /</i>	
	<i>11.1.4 Mixed-Mode Arithmetic / 11.1.5 Higher Precision</i>	
	<i>Intrinsic Functions / 11.1.6 When to Use High-Precision Real</i>	
	<i>Values / 11.1.7 Solving Large Systems of Simultaneous Linear</i>	
	<i>Equations</i>	
11.2	Alternative Lengths of the INTEGER Data Type	511
11.3	Alternative Kinds of the CHARACTER Data Type	513
11.4	The COMPLEX Data Type	514
	<i>11.4.1 Complex Constants and Variables / 11.4.2 Initializing</i>	
	<i>Complex Variables / 11.4.3 Mixed-Mode Arithmetic /</i>	
	<i>11.4.4 Using Complex Numbers with Relational</i>	
	<i>Operators / 11.4.5 COMPLEX Intrinsic Functions</i>	
11.5	Summary	524
	<i>11.5.1 Summary of Good Programming Practice /</i>	
	<i>11.5.2 Summary of Fortran Statements and Structures /</i>	
	<i>11.5.3 Exercises</i>	
12	Derived Data Types	530
12.1	Introduction to Derived Data Types	530
12.2	Working with Derived Data Types	532
12.3	Input and Output of Derived Data Types	533
12.4	Declaring Derived Data Types in Modules	534
12.5	Returning Derived Types from Functions	543
12.6	Dynamic Allocation of Derived Data Types (Fortran 2003 only)	547
12.7	Parameterized Derived Data Types (Fortran 2003 only)	548
12.8	Type Extension (Fortran 2003 only)	549
12.9	Type-Bound Procedures	550
12.10	The ASSOCIATE Construct (Fortran 2003 only)	555

12.11	Summary	556
	<i>12.11.1 Summary of Good Programming Practice /</i>	
	<i>12.11.2 Summary of Fortran Statements and Structures /</i>	
	<i>12.11.3 Exercises</i>	
13	Advanced Features of Procedures and Modules	563
13.1	Scope and Scoping Units	564
13.2	Recursive Procedures	569
13.3	Keyword Arguments and Optional Arguments	571
13.4	Procedure Interfaces and Interface Blocks	577
	<i>13.4.1 Creating Interface Blocks / 13.4.2 Notes on the Use of</i>	
	<i>Interface Blocks</i>	
13.5	Generic Procedures	581
	<i>13.5.1 User-Defined Generic Procedures / 13.5.2 Generic</i>	
	<i>Interfaces for Procedures in Modules / 13.5.3 Generic Bound</i>	
	<i>Procedures</i>	
13.6	Extending Fortran with User-Defined Operators and Assignments	594
13.7	Bound Assignments and Operators	607
13.8	Restricting Access to the Contents of a Module	608
13.9	Advanced Options of the USE Statement	611
13.10	Intrinsic Modules	615
13.11	Access to Command Line Arguments and Environment Variables	615
	<i>13.11.1 Access to Command Line Arguments /</i>	
	<i>13.11.2 Retrieving Environment Variables</i>	
13.12	The VOLATILE Attribute and Statement	618
13.13	Summary	619
	<i>13.13.1 Summary of Good Programming Practice /</i>	
	<i>13.13.2 Summary of Fortran Statements and Structures /</i>	
	<i>13.13.3 Exercises</i>	
14	Advanced I/O Concepts	633
14.1	Additional Format Descriptors	633
	<i>14.1.1 Additional Forms of the E and ES Format</i>	
	<i>Descriptors / 14.1.2 Engineering Notation—The EN</i>	
	<i>Descriptor / 14.1.3 Double-Precision Data—The</i>	
	<i>D Descriptor / 14.1.4 The Generalized (G) Format</i>	
	<i>Descriptor / 14.1.5 The Binary, Octal, and Hexadecimal (B, O,</i>	
	<i>and Z) Descriptors / 14.1.6 The TAB Descriptors / 14.1.7 The</i>	
	<i>Colon (:) Descriptor / 14.1.8 Scale Factors—The P Descriptor /</i>	
	<i>14.1.9 The SIGN Descriptors / 14.1.10 Blank Interpretation: The</i>	
	<i>BN and BZ Descriptors / 14.1.11 Rounding Control: The RU, RD,</i>	
	<i>RZ, RN, RC, and RP Descriptors (Fortran 2003 only) /</i>	

14.1.12	<i>Decimal Specifier: The DC and DP Descriptors (Fortran 2003 only)</i>	
14.2	Defaulting Values in List-Directed Input	642
14.3	Detailed Description of Fortran I/O Statements	643
14.3.1	<i>The OPEN Statement</i>	
14.3.2	<i>The CLOSE Statement</i>	
14.3.3	<i>The INQUIRE Statement</i>	
14.3.4	<i>The READ Statement</i>	
14.3.5	<i>Alternative Form of the READ Statement</i>	
14.3.6	<i>The WRITE Statement</i>	
14.3.7	<i>The PRINT Statement</i>	
14.3.8	<i>File Positioning Statements</i>	
14.3.9	<i>The ENDFILE Statement</i>	
14.3.10	<i>The WAIT Statement</i>	
14.3.11	<i>The FLUSH Statement</i>	
14.4	Namelist I/O	668
14.5	Unformatted Files	671
14.6	Direct Access Files	673
14.7	Stream Access Mode	677
14.8	Nondefault I/O for Derived Types (Fortran 2003 only)	678
14.9	Asynchronous I/O	686
14.9.1	<i>Performing Asynchronous I/O</i>	
14.9.2	<i>Problems with Asynchronous I/O</i>	
14.10	Access to Processor-Specific I/O System Information	689
14.11	Summary	690
14.11.1	<i>Summary of Good Programming Practice</i>	
14.11.2	<i>Summary of Fortran Statements and Structures</i>	
14.11.3	<i>Exercises</i>	
15	Pointers and Dynamic Data Structures	699
15.1	Pointers and Targets	700
15.1.1	<i>Pointer Assignment Statements</i>	
15.1.2	<i>Pointer Association Status</i>	
15.2	Using Pointers in Assignment Statements	706
15.3	Using Pointers with Arrays	708
15.4	Dynamic Memory Allocation with Pointers	710
15.5	Using Pointers as Components of Derived Data Types	714
15.6	Arrays of Pointers	726
15.7	Using Pointers in Procedures	728
15.7.1	<i>Using the INTENT Attribute with Pointers</i>	
15.7.2	<i>Pointer-Valued Functions</i>	
15.8	Procedure Pointers	734

15.9	Binary Tree Structures	735
	<i>15.9.1 The Significance of Binary Tree Structures /</i>	
	<i>15.9.2 Building a Binary Tree Structure</i>	
15.10	Summary	755
	<i>15.10.1 Summary of Good Programming Practice /</i>	
	<i>15.10.2 Summary of Fortran Statements and Structures /</i>	
	<i>15.10.3 Exercises</i>	
16	Object-Oriented Programming In Fortran	763
16.1	An Introduction to Object-Oriented Programming	764
	<i>16.1.1 Objects / 16.1.2 Messages / 16.1.3 Classes /</i>	
	<i>16.1.4 Class Hierarchy and Inheritance / 16.1.5 Object-Oriented Programming</i>	
16.2	The Structure of a Fortran Class	768
16.3	The CLASS Keyword	770
16.4	Implementing Classes and Objects In Fortran	772
	<i>16.4.1 Declaring Fields (Instance Variables) / 16.4.2 Creating Methods / 16.4.3 Creating (Instantiating) Objects from a Class</i>	
16.5	First Example: A timer Class	775
	<i>16.5.1 Implementing the timer Class / 16.5.2 Using the timer Class / 16.5.3 Comments on the timer Class</i>	
16.6	Categories of Methods	780
16.7	Controlling Access to Class Members	789
16.8	Finalizers	789
16.9	Inheritance and Polymorphism	793
	<i>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</i>	
16.10	Preventing Methods from Being Overridden in Subclasses	808
16.11	Abstract Classes	809
16.12	Summary	829
	<i>16.12.1 Summary of Good Programming Practice /</i>	
	<i>16.12.2 Summary of Fortran Statements and Structures /</i>	
	<i>16.12.3 Exercises</i>	
17	Redundant, Obsolescent, and Deleted Fortran Features	835
17.1	Pre-Fortran 90 Character Restrictions	836
17.2	Obsolescent Source Form	836

17.3	Redundant Data Type	837
17.4	Older, Obsolescent, and/or Undesirable Specification Statements	838
	<i>17.4.1 Pre-Fortran 90 Specification Statements /</i>	
	<i>17.4.2 The IMPLICIT Statement / 17.4.3 The DIMENSION</i>	
	<i>Statement / 17.4.4 The DATA Statement / 17.4.5 The PARAMETER</i>	
	<i>Statement</i>	
17.5	Sharing Memory Locations: COMMON and EQUIVALENCE	842
	<i>17.5.1 COMMON Blocks / 17.5.2 Initializing Data in COMMON</i>	
	<i>Blocks: The BLOCK DATA Subprogram / 17.5.3 The Unlabeled</i>	
	<i>COMMON Statement / 17.5.4 The EQUIVALENCE Statement</i>	
17.6	Undesirable Subprogram Features	848
	<i>17.6.1 Alternate Subroutine Returns / 17.6.2 Alternate Entry</i>	
	<i>Points / 17.6.3 The Statement Function / 17.6.4 Passing</i>	
	<i>Intrinsic Functions as Arguments</i>	
17.7	Miscellaneous Execution Control Features	856
	<i>17.7.1 The PAUSE Statement / 17.7.2 Arguments Associated with</i>	
	<i>the STOP Statement / 17.7.3 The END Statement</i>	
17.8	Obsolete Branching and Looping Structures	858
	<i>17.8.1 The Arithmetic IF Statement / 17.8.2 The</i>	
	<i>Unconditional GO TO Statement / 17.8.3 The Computed GO TO</i>	
	<i>Statement / 17.8.4 The Assigned GO TO Statement /</i>	
	<i>17.8.5 Older Forms of DO Loops</i>	
17.9	Redundant Features of I/O Statements	863
17.10	Summary	865
	<i>17.10.1 Summary of Good Programming Practice /</i>	
	<i>17.10.2 Summary of Fortran Statements and Structures</i>	
 Appendixes		
A.	ASCII AND EBCDIC Coding Systems	871
B.	Fortran 95/2003 Intrinsic Procedures	876
	<i>B.1. Classes of Intrinsic Procedures / B.2. Alphabetical List of</i>	
	<i>Intrinsic Procedures / B.3. Mathematical and Type Conversion</i>	
	<i>Intrinsic Procedures / B.4. Kind and Numeric Processor Intrinsic</i>	
	<i>Functions / B.5. System Environment Procedures / B.6. Bit</i>	
	<i>Intrinsic Procedures / B.7. Character Intrinsic Functions /</i>	
	<i>B.8. Array and Pointer Intrinsic Functions / B.9. Miscellaneous</i>	
	<i>Inquiry Functions / B.10. Miscellaneous Procedure</i>	
C.	Order of Statements in a Fortran 95/2003 Program	915
D.	Glossary	917
E.	Answers to Quizzes	937
Index		955

