

Preface		xiii
<b>1</b>	<b>Introduction to Computers and the Fortran Language</b>	1
1.1	The Computer	2
1.1.1	<i>The CPU / 1.1.2 Main and Secondary Memory /</i>	
1.1.3	<i>Input and Output Devices</i>	
1.2	Data Representation in a Computer	4
1.2.1	<i>The Binary Number System / 1.2.2 Octal and Hexadecimal</i>	
1.2.3	<i>Representations of Binary Numbers / 1.2.3 Types of Data Stored</i>	
1.2.3	<i>in Memory</i>	
1.3	Computer Languages	12
1.4	The History of the Fortran Language	12
1.5	The Evolution of Fortran	15
1.6	Summary	18
1.6.1	<i>Exercises</i>	
<b>2</b>	<b>Basic Elements of Fortran</b>	21
2.1	Introduction	21
2.2	The Fortran Character Set	21
2.3	The Structure of a Fortran Statement	22
2.4	The Structure of a Fortran Program	23
2.4.1	<i>The Declaration Section / 2.4.2 The Execution Section /</i>	
2.4.3	<i>The Termination Section / 2.4.4 Program Style /</i>	
2.4.5	<i>Compiling, Linking, and Executing the Fortran Program</i>	
2.5	Constants and Variables	25
2.5.1	<i>Integer Constants and Variables / 2.5.2 Real Constants</i>	
2.5.3	<i>and Variables / 2.5.3 Character Constants and Variables /</i>	
2.5.4	<i>2.5.4 Default and Explicit Variable Typing / 2.5.5 Keeping</i>	
2.5.5	<i>Constants Consistent in a Program</i>	
2.6	Assignment Statements and Arithmetic Calculations	33
2.6.1	<i>Integer Arithmetic / 2.6.2 Real Arithmetic / 2.6.3</i>	
2.6.4	<i>Hierarchy of Operations / 2.6.4 Mixed-Mode Arithmetic /</i>	
2.6.5	<i>Mixed-Mode Arithmetic and Exponentiation</i>	
2.7	Intrinsic Functions	43

	Contents
2.8	List-Directed Input and Output Statements 45
2.9	Initialization of Variables 51
2.10	The IMPLICIT NONE Statement 53
2.11	Program Examples 54
2.12	Debugging Fortran Programs 62
2.13	Summary 64
	2.13.1 Summary of Good Programming Practice /
	2.13.2 Summary of Fortran Statements / 2.13.3 Exercises
<b>3</b>	<b>Program Design and Branching Structures 76</b>
3.1	Introduction to Top-down Design Techniques 76
3.2	Use of Pseudocode and Flowcharts 81
3.3	Logical Constants, Variables, and Operators 83
	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 89
	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 112
3.6	Summary 113
	3.6.1 Summary of Good Programming Practice / 3.6.2 Summary
	of Fortran Statements and Constructs / 3.6.3 Exercises
<b>4</b>	<b>Loops and Character Manipulation 120</b>
4.1	Control Constructs: Loops 120
	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 147
	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 161
4.4	Summary 161
	4.4.1 Summary of Good Programming Practice / 4.4.2 Summary
	of Fortran Statements and Constructs / 4.4.3 Exercises

	Contents
<b>5</b>	<b>Basic I/O Concepts 173</b>
5.1	Formats and Formatted WRITE Statements 173
5.2	Output Devices 174
5.3	Format Descriptors 177
	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
5.4	Formatted READ Statements 197
	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 203
	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=
	Clause in the READ Statement / 5.5.5 File Positioning
5.6	Summary 222
	5.6.1 Summary of Good Programming Practice / 5.6.2 Summary
	of Fortran Statements and Structures / 5.6.3 Exercises
<b>6</b>	<b>Introductions to Arrays 235</b>
6.1	Declaring Arrays 236
6.2	Using Array Elements in Fortran Statements 237
	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 250
	6.3.1 Whole Array Operations / 6.3.2 Array Subsets
6.4	Input and Output 254
	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 260
6.6	When Should You Use an Array? 276

	Contents
<b>6.7</b> Summary	277
6.7.1 <i>Summary of Good Programming Practice</i> / 6.7.2 <i>Summary of Fortran Statements and Constructs</i> / 6.7.3 <i>Exercises</i>	
<b>7</b> Introduction to Procedures	285
<b>7.1</b> Subroutines	287
7.1.1 <i>Example Problem—Sorting</i> / 7.1.2 <i>The INTENT Attribute</i> / 7.1.3 <i>Variable Passing in Fortran: The Pass-by-Reference Scheme</i> / 7.1.4 <i>Passing Arrays to Subroutines</i> / 7.1.5 <i>Passing Character Variables to Subroutines</i> / 7.1.6 <i>Error Handling in Subroutines</i> / 7.1.7 <i>Examples</i>	
<b>7.2</b> Sharing Data Using Modules	307
<b>7.3</b> Module Procedures	314
<b>7.4</b> Fortran Functions	318
<b>7.5</b> Passing Procedures as Arguments to Other Procedures	325
7.5.1 <i>Passing User-Defined Functions as Arguments</i> / 7.5.2 <i>Passing Subroutines as Arguments</i>	
<b>7.6</b> Summary	329
7.6.1 <i>Summary of Good Programming Practice</i> / 7.6.2 <i>Summary of Fortran Statements and Structures</i> / 7.6.3 <i>Exercises</i>	
<b>8</b> Additional Features of Arrays	344
<b>8.1</b> Two-Dimensional or Rank-2 Arrays	344
8.1.1 <i>Declaring Rank-2 Arrays</i> / 8.1.2 <i>Rank-2 Array Storage</i> / 8.1.3 <i>Initializing Rank-2 Arrays</i> / 8.1.4 <i>Example Problem</i> / 8.1.5 <i>Whole Array Operations and Array Subsets</i>	
<b>8.2</b> Multidimensional or Rank- <i>n</i> Arrays	355
<b>8.3</b> Using Fortran Intrinsic Functions with Arrays	358
8.3.1 <i>Elemental Intrinsic Functions</i> / 8.3.2 <i>Inquiry Intrinsic Functions</i> / 8.3.3 <i>Transformational Intrinsic Functions</i>	
<b>8.4</b> Masked Array Assignment: The WHERE Construct	361
8.4.1 <i>The Fortran 90 WHERE Construct</i> / 8.4.2 <i>The Fortran 95 WHERE Construct</i> / 8.4.3 <i>The WHERE Statement</i>	
<b>8.5</b> The FORALL Construct	364
8.5.1 <i>The Form of the FORALL Construct</i> / 8.5.2 <i>The Significance of the FORALL Construct</i> / 8.5.3 <i>The FORALL Statement</i>	
<b>8.6</b> Allocatable Arrays	366
<b>8.7</b> Summary	373
8.7.1 <i>Summary of Good Programming Practice</i> / 8.7.2 <i>Summary of Fortran Statements and Constructs</i> / 8.7.3 <i>Exercises</i>	
<b>9</b> Additional Features of Procedures	384
<b>9.1</b> Passing Multidimensional Arrays to Subroutines and Functions	384

9.1.1 <i>Explicit-Shape Dummy Arrays</i> / 9.1.2 <i>Assumed-Shape Dummy Arrays</i> / 9.1.3 <i>Assumed-Size Dummy Arrays</i>	
<b>9.2</b> The SAVE Attribute and Statement	396
<b>9.3</b> Automatic Arrays	400
9.3.1 <i>Comparing Automatic Arrays and Allocatable Arrays</i> / 9.3.2 <i>Example Program</i>	
<b>9.4</b> Pure and Elemental Procedures	409
9.4.1 <i>Pure Procedures</i> / 9.4.2 <i>Elemental Procedures</i>	
<b>9.5</b> Internal Procedures	411
<b>9.6</b> Summary	413
9.6.1 <i>Summary of Good Programming Practice</i> / 9.6.2 <i>Summary of Fortran Statements and Structures</i> / 9.6.3 <i>Exercises</i>	
<b>10</b> More about Character Variables	421
<b>10.1</b> Character Comparison Operations	422
10.1.1 <i>The Relational Operators with Character Data</i> / 10.1.2 <i>The Lexical Functions LLT, LLF, LGT, and LGE</i>	
<b>10.2</b> Intrinsic Character Functions	427
<b>10.3</b> Passing Character Variables to Subroutines and Functions	430
<b>10.4</b> Variable-Length Character Functions	435
<b>10.5</b> Internal Files	438
<b>10.6</b> Example Problem	439
<b>10.7</b> Summary	444
10.7.1 <i>Summary of Good Programming Practice</i> / 10.7.2 <i>Summary of Fortran Statements and Structures</i> / 10.7.3 <i>Exercises</i>	
<b>11</b> Additional Intrinsic Data Types	450
<b>11.1</b> Alternate Kinds of the REAL Data Type	450
11.1.1 <i>Kinds of REAL Constants and Variables</i> / 11.1.2 <i>Determining the KIND of a Variable</i> / 11.1.3 <i>Selecting Precision in a Processor-Independent Manner</i> / 11.1.4 <i>Mixed-Mode Arithmetic</i> / 11.1.5 <i>Higher-Precision Intrinsic Functions</i> / 11.1.6 <i>When to Use High-Precision Real Values</i> / 11.1.7 <i>Solving Large Systems of Simultaneous Linear Equations</i>	
<b>11.2</b> Alternate Lengths of the INTEGER Data Type	473
<b>11.3</b> Alternate Kinds of the CHARACTER Data Type	475
<b>11.4</b> The COMPLEX Data Type	475
11.4.1 <i>Complex Constants and Variables</i> / 11.4.2 <i>Initializing Complex Variables</i> / 11.4.3 <i>Mixed-Mode Arithmetic</i> / 11.4.4 <i>Using Complex Numbers with Relational Operators</i> / 11.4.5 <i>COMPLEX Intrinsic Functions</i>	

	Contents
<b>11.5</b> Summary	485
<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>	
<b>12</b> Derived Data Types	490
<b>12.1</b> Introduction to Derived Data Types	490
<b>12.2</b> Working with Derived Data Types	492
<b>12.3</b> Input and Output of Derived Data Types	492
<b>12.4</b> Declaring Derived Data Types in Modules	494
<b>12.5</b> Returning Derived Types from Functions	502
<b>12.6</b> Summary	507
<i>12.6.1 Summary of Good Programming Practice /</i>	
<i>12.6.2 Summary of Fortran Statements and Structures /</i>	
<i>12.6.3 Exercises</i>	
<b>13</b> Advanced Features of Procedures and Modules	513
<b>13.1</b> Scope and Scoping Units	513
<b>13.2</b> Recursive Procedures	518
<b>13.3</b> Keyword Arguments and Optional Arguments	521
<b>13.4</b> Procedure Interfaces and Interface Blocks	526
<i>13.4.1 Creating Interface Blocks / 13.4.2 Notes on the</i>	
<i>Use of Interface Blocks</i>	
<b>13.5</b> Generic Procedures	530
<i>13.5.1 User-Defined Generic Procedures / 13.5.2 Generic</i>	
<i>Interfaces for Procedures in Modules</i>	
<b>13.6</b> Extending Fortran with User-Defined Operators and Assignments	540
<b>13.7</b> Restricting Access to the Contents of a Module	553
<b>13.8</b> Advanced Options of the USE Statement	556
<b>13.9</b> Summary	559
<i>13.9.1 Summary of Good Programming Practice /</i>	
<i>13.9.2 Summary of Fortran Statements and Structures /</i>	
<i>13.9.3 Exercises</i>	
<b>14</b> Advanced I/O Concepts	571
<b>14.1</b> Additional Format Descriptors	571
<i>14.1.1 Additional Forms of the E and ES Format Descriptors /</i>	
<i>14.1.2 Engineering Notation—The EN Descriptor /</i>	
<i>14.1.3 Double-Precision Data—The D Descriptor / 14.1.4 The</i>	
<i>Generalized (G) Format Descriptor / 14.1.5 The Binary, Octal,</i>	
<i>and Hexadecimal (B, O, and Z) Descriptors / 14.1.6 The TAB</i>	
<i>Descriptors / 14.1.7 The Colon (:) Descriptor / 14.1.8 Scale</i>	
<i>Factors—the P Descriptor / 14.1.9 The SIGN Descriptors /</i>	
<i>14.1.10 Blank Interpretation: The BN and BZ Descriptors</i>	

<b>14.2</b> Defaulting Values in List-Directed Input	578
<b>14.3</b> Detailed Description of Fortran I/O Statements	580
<i>14.3.1 The OPEN Statement / 14.3.2 The CLOSE Statement /</i>	
<i>14.3.3 The INQUIRE Statement / 14.3.4 The READ Statement /</i>	
<i>14.3.5 Alternate Form of the READ Statement / 14.3.6 The</i>	
<i>WRITE Statement / 14.3.7 The PRINT Statement / 14.3.8 File</i>	
<i>Positioning Statements / 14.3.9 The ENDFILE Statement</i>	
<b>14.4</b> Namelist I/O	598
<b>14.5</b> Unformatted Files	601
<b>14.6</b> Direct Access Files	603
<b>14.7</b> Summary	615
<i>14.7.1 Summary of Good Programming Practice /</i>	
<i>14.7.2 Summary of Fortran Statements and Structures /</i>	
<i>14.7.3 Exercises</i>	
<b>15</b> Pointers and Dynamic Data Structures	623
<b>15.1</b> Pointers and Targets	624
<i>15.1.1 Pointer Assignment Statements / 15.1.2 Pointer</i>	
<i>Association Status</i>	
<b>15.2</b> Using Pointers in Assignment Statements	630
<b>15.3</b> Using Pointers with Arrays	632
<b>15.4</b> Dynamic Memory Allocation with Pointers	634
<b>15.5</b> Using Pointers as Components of Derived Data Types	637
<b>15.6</b> Arrays of Pointers	649
<b>15.7</b> Using Pointers in Procedures	652
<b>15.8</b> Binary Tree Structures	657
<i>15.8.1 The Significance of Binary Tree Structures /</i>	
<i>15.8.2 Building a Binary Tree Structure</i>	
<b>15.9</b> Summary	677
<i>15.9.1 Summary of Good Programming Practice /</i>	
<i>15.9.2 Summary of Fortran Statements and Structures /</i>	
<i>15.9.3 Exercises</i>	
<b>16</b> Redundant, Obsolescent, and Deleted FORTRAN Features	684
<b>16.1</b> Pre-Fortran 90 Character Restrictions	684
<b>16.2</b> Obsolescent Source Form	685
<b>16.3</b> Redundant Data Type	686
<b>16.4</b> Older, Obsolescent, and Undesirable Specification Statements	687
<i>16.4.1 Pre-Fortran 90 Specification Statements / 16.4.2 The</i>	
<i>IMPLICIT Statement / 16.4.3 The DIMENSION Statement /</i>	
<i>16.4.4 The DATA Statement / 16.4.5 The PARAMETER</i>	
<i>Statement</i>	

<b>16.5</b>	Sharing Memory Locations: COMMON and EQUIVALENCE	690
	<i>16.5.1 COMMON Blocks / 16.5.2 Initializing Data in COMMON Blocks: The BLOCK DATA Subprogram / 16.5.3 The Unlabeled COMMON Statement / 16.5.4 The EQUIVALENCE Statement</i>	
<b>16.6</b>	Undesirable Subprogram Features	697
	<i>16.6.1 Alternate Subroutine Returns / 16.6.2 Alternate Entry Points / 16.6.3 The Statement Function / 16.6.4 Passing Intrinsic Functions as Arguments</i>	
<b>16.7</b>	Miscellaneous Execution Control Features	704
	<i>16.7.1 The PAUSE Statement / 16.7.2 Arguments Associated with the STOP Statement / 16.7.3 The END Statement</i>	
<b>16.8</b>	Obsolete Branching and Looping Structures	707
	<i>16.8.1 The Arithmetic IF Statement / 16.8.2 The Unconditional GO TO Statement / 16.8.3 The Computed GO TO Statement / 16.8.4 The Assigned GO TO Statement / 16.8.5 Older Forms of DO Loops</i>	
<b>16.9</b>	Redundant Features of I/O Statements	711
<b>16.10</b>	Summary	712
	<i>16.10.1 Summary of Good Programming Practice / 16.10.2 Summary of Fortran Statements and Structures</i>	
<b>Appendixes</b>		
<b>A</b>	ASCII and EBCDIC Coding Systems	719
<b>B</b>	Fortran 90/95 Intrinsic Procedures	725
	<i>B.1 Classes of Intrinsic Procedures / B.2 Alphabetical List of Intrinsic Procedures / B.3 Mathematical and Type Conversion Intrinsic Procedures / B.4 Kind and Numeric Processor Intrinsic Functions / B.5 Date and Time Intrinsic Subroutines / B.6 Bit Intrinsic Procedures / B.7 Character Intrinsic Functions / B.8 Array and Pointer Intrinsic Functions / B.9 Miscellaneous Inquiry Functions</i>	
<b>C</b>	Order of Statements in a Fortran 90/95 Program	759
<b>D</b>	Glossary	761
<b>E</b>	Answers to Quizzes	779
	<b>Index</b>	795