Modern Fortran
Style and Usage

Fortran is one of the oldest high-level languages and remains the premier language for writing code for science and engineering applications. Modern Fortran: Style and Usage is a book for anyone who uses Fortran, from the novice learner to the advanced expert. It describes best practices for programmers, scientists, engineers, computer scientists, and researchers who want to apply good style and incorporate rigorous usage in their own Fortran code or to establish guidelines for a team project. The presentation concentrates primarily on the characteristics of Fortran 2003, while also describing methods in Fortran 90/95 and valuable new features in Fortran 2008.

The authors draw on more than a half century of experience writing production Fortran code to present clear succinct guidelines on formatting, naming, documenting, programming, and packaging conventions and various programming paradigms such as parallel processing (including OpenMP, MPI, and coarrays), OOP, generic programming, and C language interoperability. Programmers working with legacy code will especially appreciate the section on updating old programs.

Norman S. Clerman is currently a private consultant. He was formerly the Chief Computer Scientist at Opcon Design Associates, LLC, a small company engaged in lens design.

Walter Spector has been employed by Silicon Graphics International’s Professional Services organization (formerly Cray Research Inc.) since 1984.
PRELIMINARY REPORT

Programming Research Group
Applied Science Division
International Business Machines Corporation
November 10, 1954

Specifications for
The IBM Mathematical FORMULA TRANSLATING System,

Copyright, 1954, by International Business Machines Corporation
590 Madison Avenue, New York, 22, New York

Reproduced with permission of IBM Corporate Archives
Modern Fortran

Style and Usage

Norman S. Clerman
Walter Spector

Silicon Graphics, Inc., Fremont, California
To Miriam for all her love and endless encouragement.
Norman S. Clerman

To Irene, Nyssa, Simon, and Sammy with love.
Walter Spector
Contents

Source Code Listings .............................................. page xi
Preface ........................................................ xiii
 Audience ......................................................... xiii
 General Considerations ......................................... xiii
 Organization .................................................... xiv

1. Introduction .................................................. 1
  1.1 Acknowledgments ............................................ 1
  1.2 Typographical Conventions .................................... 1
  1.3 Source Code Listings .......................................... 2

2. General Principles ........................................... 3

3. Formatting Conventions ....................................... 9
  3.1 Source Form ................................................. 9
  3.2 Case ....................................................... 10
  3.3 Indentation ................................................. 13
  3.4 White Space ................................................ 17
  3.5 Statements and Expressions ................................... 20
  3.6 Symbols and Numbers ....................................... 22

4. Naming Conventions ......................................... 24
  4.1 Modules, Procedures, and Types ............................... 24
  4.2 Supplemental Notation ....................................... 27
  4.3 Data Objects ................................................ 30
  4.4 Construct Blocks ............................................ 34

5. Documentation Conventions ............................. 38

6. Programming Principles ................................... 44
  6.1 Standards Conformance ...................................... 44
  6.2 Module Design .............................................. 44
  6.3 Procedure Arguments ........................................ 50
  6.4 Argument Verification ....................................... 82
  6.5 Recursive Procedure Design ................................... 86
  6.6 State, Side Effects, and Thread Safety .................... 90
7. Programming Conventions .................................................. 98
   7.1 Declaration and Initialization of Data Objects ................... 98
   7.2 Allocatable and Pointer Attributes ............................... 102
   7.3 Dynamic Data Attributes ....................................... 128
   7.4 Control Flow .................................................. 133
   7.5 Character String Expressions ................................... 138

8. Input and Output ........................................................... 140
   8.1 General I/O Operations ......................................... 140
   8.2 Formatted, Unformatted, and Stream I/O ....................... 155
   8.3 Messages ....................................................... 163

9. Packaging Conventions ................................................... 165
   9.1 Files .......................................................... 165
   9.2 External Procedures ............................................ 165
   9.3 Submodules .................................................... 167

10. Generic Programming .................................................... 171
    10.1 Generic Code Techniques .................................... 172
    10.2 Preprocessor Replication .................................... 179

11. Object Orientation ....................................................... 183
    11.1 Type and Class Design ....................................... 184
    11.2 Creation, Construction, Initialization, and Assignment .... 189
    11.3 Type-Bound Procedures and Procedure Pointer Components . 197
    11.4 Inheritance .................................................. 208
    11.5 Polymorphism ................................................ 212

12. Parallel Processing ...................................................... 216
    12.1 Preliminary Considerations .................................. 217
    12.2 Parallel Program Patterns ................................... 220
    12.3 Fortran 2008 Parallel Processing ............................ 242

13. Numerical Types, Kinds, and Operations .............................. 247
    13.1 The Concept of KIND ......................................... 247
    13.2 Floating-Point Operations ................................... 249
    13.3 Floating-Point Exceptions ................................... 253
    13.4 Bit Manipulation ............................................. 258

14. C Interoperability ......................................................... 262

15. Updating Old Programs ................................................... 271
    15.1 Statements .................................................... 271
    15.2 Techniques .................................................... 282
    15.3 Nonstandard Features ......................................... 284
CONTENTS ix

A. Source Code ........................................... 289
   A.1 Type_bound_demo ......................................... 289
   A.2 Unlimited_demo ........................................... 295

B. Rule List ................................................. 308

Bibliography ................................................ 321
Index .............................................................. 327
Source Code Listings

6.1 Program Intent_exp_imp ......................................... 50
6.2 Program Intent_out_test .......................................... 57
6.3 Program Point_intent_in_out ..................................... 67
6.4 Program Arg_par_test ............................................. 83
7.1 Program Return_string ............................................ 111
7.2 Module Global_target_mod ....................................... 123
7.3 Program Proc_point_test .......................................... 126
7.4 Function To_lower ................................................. 138
8.1 Module Lun_mod ................................................... 141
8.2 Subroutine Process_elev_file .................................... 144
8.3 Program DTIO_demo ............................................... 148
8.4 Program Dbuf_example ............................................ 152
8.5 Program Read_sensor_file ....................................... 157
8.6 Program Stream_demo ............................................. 161
10.1 Module My_type_mod ............................................. 176
10.2 Module Scan_mod ................................................ 181
11.1 Program Comp_name ............................................. 191
11.2 Program Transaction ............................................. 201
12.1 Program Shift_demo .............................................. 224
12.2 Program MPI_hello ............................................... 234
12.3 Program Comm_random .......................................... 236
12.4 Module Comm_mod ............................................... 239
12.5 Program Coarray_random ....................................... 244
13.1 Function Near0_dp .............................................. 250
13.2 Function Divide_DP ........................................ 255
14.1 Program Pid_printer .................................... 263
14.2 Subroutine Fortran_FFT ................................. 264
14.3 Program c_f_pointer_test .............................. 265
14.4 Program Char_test .................................... 267
A.1 Program Type_bound_demo .............................. 289
A.2 Program Unlimited_demo ............................... 295
Preface

Audience
We wrote this book for anyone who writes programs using Fortran. We think it will be useful for the following categories of programmers:

- Those who are learning Fortran from scratch and want to start on the right foot.
- Those who are familiar with pre-modern FORTRAN (up to FORTAN 77) and would like to learn some of the new concepts and techniques of modern Fortran (Fortran 90 to Fortran 2008).
- Those who have advanced knowledge of Fortran, have experimented with various styles, and are open to new ways to improve their programs.
- Those who have experience with other languages, who know the importance of good coding style and who want to apply it to their Fortran code.
- Those who want to create coding guidelines for teams of programmers, or establish a good style for a project.

General Considerations
The past four revisions of the Fortran Standard comprise alternating major and minor revisions, Fortran 90 and Fortran 2003 being the former, Fortran 95 and Fortran 2008 the latter. This book concentrates primarily on Fortran 2003. We present Fortran 90/95 methods and note where the techniques and methods of Fortran 2003 supercede them. Fortran 2008 capabilities are described, but to a lesser extent.

Each programmer will judge the importance of the new features of the language based on his or her experience and needs. The new C Interoperability may be very important to a programmer who often needs to build an application written in both languages. To another, the new object orientation may be more important, and for a third, the new parallel programming features in Fortran 2008 will be of great value.

This book is not a textbook, nor is it an exhaustive guide to the language. For those needs, in addition to the standard itself, we refer the reader to References [55] and [1], both with long publishing histories. This book is, instead, a book of guidelines, or rules, that a reader may want to adopt when writing code. We attempt to explain what we consider to be good coding techniques. Some of these are general – they apply to all languages – others are specific to Fortran. We
concentrate on characteristics of the language, especially those of Fortran 2003, that we feel require careful and detailed explanation. Others we discuss only briefly or do not discuss at all. We have striven to write each guideline in a clear and succinct manner, brevity being a key goal. We advocate strict adherence to some of the rules. It serves no purpose, for example, not to include an `implicit none` statement in every program unit, and the advantages of doing so are enormous. Our approach to other rules, though, is more to suggest coding or style guidelines than to dictate them, and, in some instances, simply to offer alternatives. In each rule, we make frequent references to other rules, while at the same time trying to make each independent so the reader does not have to constantly jump from one to another. All the same, those guidelines that present new and somewhat complicated aspects of the language, by necessity, require more space. We have taken “typographical license” and somewhat violated our own formatting rules. For instance, we use more end-of-line comments than we recommend for your code.

The next section outlines the organization of the book, listing the subjects covered in each chapter. The earlier chapters cover basic topics that apply to all languages; the later chapters deal more specifically with Fortran. In the examples and the sample code, there is no natural progression from simple to more complex code. The assumption is that the reader is at least somewhat familiar with Fortran. The book contains code utilizing new language features that are explained later in the book. We believe that cross references to other sections and other rules will suffice to make these clear.

Organization

Chapter 1 defines the typographical conventions used throughout our book. The next several chapters take a “top-down” approach to organizing and coding Fortran applications. Several design objectives that constitute goals for the writing of excellent code are presented in Chapter 2.

Chapter 3 discusses source form and layout. Chapter 4 is dedicated to the naming of various entities. Chapter 5 discusses comments and internal program documentation.

Chapter 6 begins to discuss specifics of using modules as a key to organizing code, the use of derived types, and organization of procedures and their argument lists. Chapter 7 discusses data in more detail, and control flow through procedures. Chapter 8 presents input/output (I/O). Chapter 9 provides more on the organization of modules and program units and their packaging in files. Its final section covers an important new Fortran 2008 feature for large programs – submodules. Submodules offer a solution to the well-known problem of “compilation cascades.” The remaining chapters are a potpourri of special topics.

Chapter 10 explains several techniques for writing generic code. Chapter 11 forms an introduction to object-oriented programming, a major feature of Fortran 2003.
Chapter 12 is an introduction to several forms of parallel processing that are commonly found in programs. Chapter 13 contains guidelines about the considerations the programmer must make when writing code that performs floating-point numerical operations. Chapter 14 is an introduction to C Interoperability.

The final chapter, Chapter 15, contains recommendations on updating programs that pre-date Fortran 90.

These serve as starting points for the interested programmer. Far more extensive and detailed references are available for most of the topics (References [1], [55], and [39]).

Appendix A contains the source code listing of programs Type_bound_demo and Unlimited_demo, two complete programs that are referenced from several different chapters in the book, and Appendix B collects all the rules in a list with the page reference to their location.

References to the Fortran Standard refer to Fortran 2003, unless otherwise specified (see Reference [39]).