1 Overview

“I don't know what the language of the year 2000 will look like, but it will be called Fortran.”

C.A.R. Hoare

Aims

The aims of the chapter are to provide a background to the organisation of the book.

1 Overview

The book aims to provide coverage of a recommended subset of the full Fortran language. The subset we have chosen is one that fits most closely with the theory and practice of structured programming, data structuring and software engineering.

This book has been written for both complete beginners with little or no programming background and experienced Fortran programmers who want to update their skills and move to a modern version of the language.

Chapters 2–4 provide a short background to computer systems and their use:

- Chapter 2 looks at the basics of computer systems from the hardware point of view.
- Chapter 3 provides a short history of operating system developments and looks at some commonly used operating systems.
- Chapter 4 looks at some of the fundamentals of using a computer system.

These three chapters provide information that will be very helpful in the longer term for the successful use of computer systems for programming.

Chapters 5 and 6 provide a coverage of problem solving and the history and development of programming languages. Chapter 5 is essential for the beginner as the concepts introduced there are used and expanded on throughout the rest of the book. Chapter 6 must be read at some point but can be omitted initially. Programming languages evolve and some understanding of where Fortran has come from and where it is going will prove valuable in the longer term:

- Chapter 5 looks at problem solving in some depth, and there is a coverage of the way we define problems, the role of algorithms, the use of both top-down and bottom-up methods, and the requirement for formal systems analysis and design for more complex problems.
- Chapter 6 looks at the history and development of programming languages. This is essential as Fortran has evolved considerably from its origins in the mid-1950s, through the first standard in 1966, the Fortran 77 standard, the Fortran 90 standard, the Fortran 95 standard, TR 15580 and TR 15581, Fortran 2003 and beyond. It helps to put many of the current and proposed features of Fortran into context. Languages covered include Cobol, Algol, Lisp, Snobol, PL/1, Algol 68, Simula, Pascal, APL, Basic, C, Ada, Modula, Modula 2, Logo, Prolog, SQL, ICON, Oberon, Oberon 2, Smalltalk, C++, C# and Java.

Chapters 7 through 11 cover the major features provided in Fortran for numeric programming in the first instance and for general purpose programming in the second. Each chapter has a set of problems. It is essential that a reasonable range of problems is attempted and completed, as it is impossible to learn any language without practice:

- Chapter 7 provides an introduction to programming with some simple Fortran examples. For people with a knowledge of programming this chapter can be covered fairly quickly.
- Chapter 8 looks at arithmetic in some depth, with a coverage of the various numeric data types, expressions and assignment of scalar variables. There is also a thorough coverage of the facilities provided in Fortran to help write programs that work on different hardware platforms.
- Chapter 9 is an introduction to arrays and DO loops. The chapter starts with some examples of tabular structures that one should be familiar with. There is then an examination of what concepts we need in a programming language to support manipulation of tabular data.
- Chapter 10 takes the ideas introduced in chapters 8 and 9 and extends them to higher-dimensioned arrays, additional forms of the DIMENSION attribute and corresponding form of the DO loop, and the use of looping for the control of repetition and manipulation of tabular information without the use of arrays.
- Chapter 11 looks at more of the facilities offered for the manipulation of whole arrays and array sections, ways in which we can initialise arrays using constructors, look more formally at the concepts we need to be able to accurately describe and understand arrays, and finally look at the differences between the way Fortran allows us to use arrays and the mathematical rules governing matrices.

Chapters 9 through 11 provide a coverage of some of the more important features and uses of arrays in the field of numerical problem solving. The framework provided here is drawn upon in later chapters in the book with more complex and realistic examples.

Chapters 12, 13 and 14 look at input and output (I/O) and file handling in Fortran. An understanding of I/O is necessary for the development of so-called production, non interactive programs. These are essentially fully developed programs that are used repeatedly with a variety of data inputs and results:

- Chapter 12 looks at output of results and how to generate something that is more comprehensible and easy to read than what is available with free format output and also how to write the results to a file rather than the screen.

- Chapter 13 extends the ideas introduced in Chapter 12 on output to cover input of data, or reading data into a program and also considers file I/O.
- Chapter 14 provides a coverage of files.

Chapter 15 introduces the first building block available in Fortran for the construction of programs for the solution of larger, more complex problems. It looks at the functions available in Fortran, the so-called intrinsic functions and procedures (over 100 of them) and covers how you can define and use your own functions.

It is essential to develop an understanding of the functions provided by the language and when it is necessary to write your own.

Chapter 16 introduces more formally the concept of control structures and their role in structured programming. Some of the control structures available in Fortran are introduced in earlier chapters, but there is a summary here of those already covered plus several new ones that complete our coverage of a minimal working set.

Chapters 17 through 21 complete our coverage of the facilities for data typing and structuring provided by Fortran, both predefined and user defined. Fortran has now caught up with some of the major developments in the data-structuring area of the last 20 years, which have been available in other languages for some time:

- Chapter 17 looks at the character data type in Fortran. There is a coverage of I/O again, with the operators available — only one in fact.
- Chapter 18 looks at the last numeric data type in Fortran, the complex data type. This data type is essential to the solution of a small class of problems in mathematics and engineering.
- Chapter 19 looks at the logical data type. The material covered here helps considerably in increasing the power and sophistication of the way we use and construct logical expressions in Fortran. This proves invaluable in the construction and use of logical expressions in control structures.
- Chapters 20 looks at user-defined data types. This introduces another major new feature of Fortran. Previous versions of the language lacked any facilities in this area. This meant that in many applications earlier versions of Fortran were not the language of first choice for many people.
- Chapter 21 looks at the dynamic data-structuring facilities now available in Fortran. Examples are drawn from a range of sources.

These chapters conclude coverage of the data-structuring facilities provided by Fortran. There are problems that will require facilities not provided, but it is surprising what can be achieved with the set now provided in Fortran. The material covered is extended into more realistic examples when we look at the construction of larger and more complex programs in the last few chapters in the book.

The next two chapters look at the second major building block in Fortran — the subroutine. Chapter 22 provides a gentle introduction to some of the fundamental concepts of subroutine definition and use and Chapter 23 extends these ideas.

Chapter 24 introduces the concept of a module and the range of things that it brings to Fortran.

Chapter 25 looks at converting to modern Fortran. A number of examples are used and several software tools are examined.

Chapter 26 has a number of case studies helping to pull together the ideas presented in the earlier chapters.

Chapter 27 looks at ISO TR 15580 — IEEE Arithmetic.

Chapter 28 deals with ISO TR 15581 — Allocatable Enhancements

Chapters 29 covers the new features of Fortran 2003 and ISO/IEC DTR 19767, Enhanced Module Facilities.

Chapter 30 examines parallel Fortran.

Chapter 31 ties up some loose ends. It looks at program development and software engineering, modules, programming style, data structures, algorithms, structured programming, recursion and recursion removal, efficiency in space and time, program testing, simple debugging techniques, software tools and numerical software sources. There is also coverage of the various internet resources available for Fortran.

Many of the chapters have annotated bibliographies. These often have pointers and directions for further reading. The coverage provided cannot be seen in isolation. The concepts introduced are by intention brief, and fuller coverage must be sought where necessary.

There are several appendices:

- Appendix A — This is a glossary which provides coverage of both the new concepts provided by Fortran and a range of computing terms and ideas.
- Appendix B — Provides an example of a simple program in a number of the languages described in the chapter on program language development. There is also coverage of the standards that apply.
- Appendix C — The ASCII character set.

- Appendix D — Contains a list of all of the intrinsic procedures in Fortran and includes a full explanation of each procedure with a coverage of the rules and restrictions that apply and examples of use.
- Appendix E — Contains the English and Latin text extracts used in one of the problems in the chapter on characters.
- Appendix F — Contains the coded text extract used in one of the problems in Chapter 17.
- Appendix G — Formal syntax
- Appendix H — Sample compiler options

This book is not and cannot possibly be completely self-contained and exhaustive in its coverage of the Fortran language. Our first intention has been to produce a coverage of the features that will get you started with Fortran and enable you to solve quite a wide range of problems successfully.

Fortran, like most languages, has features that are of relatively little use or make the construction of larger-scale programs more difficult, especially when moving between hardware platforms. We have deliberately avoided these features.

Another problem is backwards compatibility with Fortran 77. Existing Fortran 77 programs have to be maintained, and there is much in that language that is *deprecated* or *obsolescent* in terms of Fortran 95 and Fortran 2003.

We have aimed to introduce a working subset of the new language that emphasises the better constructs provided in Fortran over its predecessors, Fortran 77 and Fortran 66.

All in all Fortran is an exciting language, and it has *caught up* with language developments of the 1970s, 1980s, and 1990s.

A range of hardware platforms, operating systems and Fortran compilers were used. These include:

- DEC VAX under VMS and later Open VMS using the NAG Fortran 90 compiler.
- DEC Alpha under Open VMS using the DEC/Compaq Fortran 90 compiler.
- PC under DOS and Windows, DEC/Compaq Fortran 90.
- PC under DOS and Windows, DEC/Compaq/HP Fortran 95.
- PC under DOS and Windows, NAG/Salford Fortran 90.
- PC under DOS and Windows, Lahey Fujitsu Fortran 95 PRO 5.7.

- PC under DOS and Windows, Intel.
- PC under DOS and Windows, NAGWare f95.
- Sun UltraSparc under Solaris using NAGWare F90.
- Sun UltraSparc under Solaris using NAGACE F90.
- Sun UltraSparc under Solaris using NAGWare F95.
- Sun UltraSparc under Solaris using Sun F90.
- Intel Linux, NAGWare f95.
- Intel Linux, Lahey Fujitsu Fortran 95 PRO, 6.1.
- Intel Linux, Intel.

Our recommendation is that you use at least two compilers in the development of your code. Moving code between platforms teaches you a lot.

We are the current owners of the Fortran 90 list, and quoting the introduction "*This list covers all aspects of Fortran 90 and HPF, the new standard(s) for Fortran. The emphasis should be on the *new* features of Fortran 90. It welcomes contributions from people who write Fortran 90 applications, teach it in courses, want to port programs and use it on (super)computers.*"

Visit:

- http://www.jiscmail.ac.uk/lists/comp-fortran-90.html

for more information.

Ian Chivers is also Editor of Fortran Forum, the SIGPLAN Special Interest Publication on Fortran, ACM Press.