

# Chapter 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.

### 1.1 Introduction

The book aims to provide coverage of a reasonable working subset of the Fortran programming language. The subset chosen should enable you to solve quite a wide range of frequently occurring problems.

This book has been written for three audiences:

- the complete beginner with little or no programming background
- an experienced Fortran programmer who wants to update their skills and move to a modern version of the language
- a programmer familiar with another language wanting to see what modern Fortran has to offer

Chapters 2 and 3 provide a coverage of problem solving and the history and development of programming languages. Chapter 2 is essential for the beginner as the concepts introduced there are used and expanded on throughout the rest of the book. Chapter 3 should 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 2 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 3 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, Fortran 2008 to Fortran 2018. 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#, Java and Python.

Chapters 4–8 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 are attempted and completed, as it is impossible to learn any language without practice.

- Chapter 4 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 5 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 6 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 7 takes the ideas introduced in Chap. 6 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 8 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, 10 and 11 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 9 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 10 extends the ideas introduced in Chap. 9 to cover input of data, or reading data into a program and also considers file I/O.

- Chapter 11 provides a summary of input and output concepts introduced in Chaps. 9 and 10, and expands on them by introducing additional features of the read, write, open and close statements.

Chapter 12 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 13 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 14–16 complete our coverage of the intrinsic facilities in Fortran for data typing.

- Chapter 14 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 15 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 16 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.

Chapter 17 introduces derived or user defined types with a small number of examples.

Chapter 18 looks at the dynamic data-structuring facilities now available in Fortran with the addition of pointers. This chapter looks at the basic syntax of pointers. They are used in range of examples in later chapters in the book.

The next two chapters look at the second major building block in Fortran — the subroutine. Chapter 19 provides a gentle introduction to some of the fundamental concepts of subroutine definition and use and Chapter 20 extends these ideas.

Chapter 21 introduces one of modern Fortran's major key features - the module. A Fortran module can be thought of as equivalent to a class in C++, Java and C#. This chapter looks at the basic syntax, with a couple of simple examples.

Chapter 22 looks at simple data structuring in Fortran, as we have now covered modules in a bit more depth.

Chapter 23 introduces algorithms and the big O notation.

Chapter 24 looks briefly at operator overloading, first introduced in Fortran 90.

Chapter 25 looks at generic programming.

Chapter 26 has a small set of mathematical examples.

Chapter 27 introduces parameterised derived types.

Chapter 28 introduces object oriented programming in Fortran.

Chapter 29 is the second chapter on object oriented programming  
Chapters 30–34 look at parallel programming in Fortran with coverage of MPI, OpenMP and Coarray Fortran.

Chapter 35 looks at C interoperability.

Chapter 36 looks at IEEE Arithmetic support in Fortran.

Chapter 37 looks at derived type I/O in Fortran

Chapter 38 looks at a number examples of sorting and searching

Chapter 39 looks at handling missing data in calculations

Chapter 40 looks at converting from Fortran 77 to more modern Fortran.

Chapter 41 looks at using a graphics library for plotting

Chapter 42 has an example of abstract interfaces and procedure pointers in Fortran

Some 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. References to the standard in the book are to the current Fortran 2018 revision unless otherwise stated. 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—is a reference appendix on attribute declarations and specifications
- Appendix C—provides details of compatibility between standards
- Appendix D—Contains a list of some of the more commonly used intrinsic procedures in Fortran and includes an explanation of each procedure with a coverage of the rules and restrictions that apply and examples of use where appropriate. There also some tables summarising information about the procedures
- Appendix E—Contains the English and Latin text extracts used in one of the problems in the chapter on characters, and the coded text extract used in one of the problems in Chap. 14.
- Appendix F—Formal syntax.
- Appendix G—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 a range of problems successfully. All in all Fortran is an exciting language, and it has caught up with language developments of the last 50 years.

## 1.2 Program Examples

All of the program examples are available on line at

<https://www.fortranplus.co.uk/>

All examples have been reformatted using the Nag compiler *polish* option. This makes the programs have a consistent style. The examples in the book have been formatted to have a line length of 48 characters to fit the printed page. They were then manually edited to improve where the lines broke. The examples on the web site have been formatted to have a line length of 132 characters.

### **1.3 Web Addresses**

Web addresses are used throughout the book. As some of these are likely to change over the lifetime of the book our web site will have up to date addresses. We have organised them by chapter.