2 Fortran Concepts and Terms

- A **Program** is an organized collection of program units. There must be exactly one main program, and in addition there may be modules, external subprograms, and block data units. Elements described by means other than Fortran may be included.

- A **Module** provides a means of packaging related data and procedures, and hiding information not needed outside the module. There are several intrinsic modules.

- The **Data Environment** consists of the data objects upon which operations will be performed to create desired results or values. These objects may have declared and dynamic types; they may have type parameters, and they may possess attributes such as dimensionality. They need not exist for the whole execution of the program. Allocatable objects and pointer targets may be created when needed and released when no longer needed.

- **Program Execution** begins with the first executable construct in the main program and continues with successive constructs unless there is a change in the flow of control. When a procedure is invoked, its execution begins with its first executable construct. On normal return, execution continues where it left off. Execution may occur simultaneously with input/output processes.

- The **Definition Status** of a variable indicates whether or not the variable has a value; the value may change during execution. Most variables are initially undefined and become defined when they acquire a value. The status also may become undefined during execution. Pointers have both an association status and a definition status. Allocatable objects have both an allocation status and a definition status.

- **Scope** and **Association** determine where and by what names various entities are known and accessible in a program. These concepts form the information backbone of the language.

This chapter introduces the basic concepts and fundamental terms needed to understand Fortran. Some terms are defined implicitly by the syntax rules. Others, such as “associated” or “present” are ordinary English words, but they have a specific Fortran meaning.

One of the major concepts involves the organization of a program. A program consists of program units; program units consist of Fortran statements. Some statements are executable; some are not. In general, the nonexecutable statements define the data environment, and the executable statements specify the actions taken. This chapter presents the high-level syntax rules for a Fortran program. It also describes the order in which constructs and statements may appear in a program and concludes with an example of a short, but complete, Fortran program.
While there is some discussion of language features here to help explain various terms and concepts, Chapters 3–16 contain the complete description of all language features.

2.1 Program Organization

A collection of program units constitutes an executable program. A Fortran program must have one main program and may have any number of the other program units. Program units may serve as hosts for smaller scoping units. Information may be hidden within part of a program or communicated to other parts of a program by various means. The programmer may control the parts of a program in which information is accessible.

With the introduction of C interoperability in Fortran 2003, it is possible to include, with much greater ease and portability, external procedures and other entities defined by a means other than Fortran. A processor has one or more companion processors. A companion processor is a processor-dependent mechanism by which global data and procedures may be referenced or defined. It may be the Fortran processor itself, or it may be another Fortran processor. If a procedure is defined by means of a companion processor that is not the Fortran processor itself, the standard refers to the C function that defines the procedure. Although the procedure need not be defined by means of the C programming language, the interoperability mechanisms are designed to mesh well with C.

2.1.1 Program Units

A Fortran program unit is one of the following:

- main program
- module
- external subprogram
- block data

A Fortran program may consist of only a main program, although usually there are also modules and/or external subprograms which may be subroutine or function subprograms. These program units contain constructs and statements that define the data environment and the steps necessary to perform calculations. Each program unit has an END statement to terminate the program unit. Each has a special initial statement as well, but the initial statement for a main program is optional. For example, a program might contain a main program, a module, and a subroutine:

```
program task
   . . .
call calc (z)
   . . .
end program task
```