8 Block Constructs and Execution Control

- A **Block** is a bounded sequence of executable constructs and statements that is treated as a unit. It may be empty.

- A **Block Construct** has an initial statement and a terminal statement; it contains zero or more blocks and the statements that bound the blocks. It is used to control execution or simply to define a region of code.

- The **ASSOCIATE Construct** allows a named entity, the associate name, to be associated with a variable or expression during the execution of its single block.

- The **IF Construct** contains one or more blocks; at most one is chosen for execution. The choice is based on the value of a logical expression.

- The **CASE Construct** contains zero or more blocks; at most one is selected for execution. The selection is based on the value of an integer, character, or logical expression.

- The **SELECT TYPE Construct** contains zero or more blocks; at most one is selected for execution. Rather than a value, as in the CASE construct, the selection is based on the dynamic type of a variable or expression.

- The **DO Construct** contains a single block that is executed repeatedly. There are multiple forms for controlling the execution. A **CYCLE statement** is permitted at any point to start the next execution of the block. An **EXIT statement** terminates the repetition.

- The **IF Statement** permits the execution of a single statement if the contained logical expression evaluates to true.

- The **GO TO Statement** transfers control to a labeled statement.

- The **CONTINUE Statement** has no effect on execution.

- The **STOP Statement** causes termination of the execution of the program.

- The **Computed GO TO Statement**, the **Arithmetic IF Statement**, and the **nonblock DO** are obsolescent features that use labels.

This chapter describes five block constructs, four of which are execution control constructs. It also describes the executable statements that are used to alter the normal execution sequence. The block constructs that are control constructs are the IF construct, the CASE construct, the SELECT TYPE construct, and the DO construct. Individual statements that alter the normal execution sequence include the EXIT and CYCLE statements that are special statements for DO constructs, branch statements such as the
GO TO statement, and a statement that causes execution to cease, the STOP statement. The fifth block construct described in this chapter is the ASSOCIATE construct. Its single block defines a region of the program in which an associate name may be used instead of a longer or more complicated variable or expression.

There are two other constructs that look like control constructs, but are really forms of assignment. These are the WHERE construct (7.5.6), which somewhat resembles an IF construct and the FORALL construct (7.5.7), which somewhat resembles a DO construct.

With any of the block constructs, construct names may be used to identify the constructs and also to identify which DO constructs, particularly in a nest of DO constructs, are being terminated or cycled when using the EXIT or CYCLE statements.

### 8.1 Blocks and Construct Names

A **block** (R801) has the form:

```
[ execution-part-construct ] ...
```

A block is treated as a whole. Not every statement or construct in a block need be executed; for example, a branch statement early in the block may prevent subsequent statements in the block from being executed. This is still a complete execution of the block.

A control construct consists of zero or more blocks and the control logic that explicitly or implicitly encloses these blocks. A construct has an initial statement and a terminal statement. In constructs that have more than one block, there are additional statements between blocks that determine which block is chosen for execution. The control for the DO construct determines how many times its block will be executed. An example of a named executable construct controlling a block of statements is:

```
INNER: IF (I<=1) THEN  ! Initial statement of the IF construct
    X = 1.2*I            ! First statement of the block
    Y = COS(X)          ! Final statement of the block
END IF INNER          ! Terminal statement of the IF construct
```

All of the block constructs (ASSOCIATE, CASE, DO, IF, and SELECT TYPE), as well as the FORALL and WHERE constructs, may have construct names. If a construct name is used, it must appear on the initial statement of the construct and a matching occurrence of the same name must appear on the terminal statement of the construct. If there is no construct name on the initial statement, the terminal statement must not have a construct name. If one of the internal control statements contains a construct name, it must be the same name as the one on the initial and terminal statements. The same construct name must not be used for different constructs in the same scoping unit.