CHAPTER 4

The Concept of Action

Essential to a computer program is action. That is, a program must do something with its data — even if that action is the choice of doing nothing! Statements describe these actions. Statements are either simple (e.g., the assignment statement) or structured. See the syntax diagram for Statement (Figure 4.a).

4.A. The Assignment Statement and Expressions

The most fundamental of statements is the assignment statement. It specifies that a newly computed value, specified by an expression, be assigned to a variable. Assignment statements have the form shown in Figure 4.b. The := symbol denotes assignment and is not to be confused with the relational operator =. The statement “\(A := 5\)” is read “the current value of \(A\) is replaced with the value 5,” or simply, “\(A\) becomes 5.”

A variable (see Figure 4.c) may be an entire variable representing all the data storage for a simple, structured, or pointer type. In the case of structured types (see Chapters 6 through 9), a variable may be a component variable or a buffer variable representing one component of the data storage. For pointer types, a variable may be an identified variable representing data storage indirectly referenced by a pointer.

An expression consists of operators and operands. An operand may be a constant, variable, array-parameter bound (discussed in Chapter 28).
11), or function designator. (A function designator specifies activation of a function. Predeclared functions are listed in Appendix A; user-declared functions are explained in Chapter 11.)

![Figure 4.a Syntax diagram for Statement](image)

**Figure 4.a** Syntax diagram for *Statement*

![Figure 4.b Syntax diagram for AssignmentStatement](image)

**Figure 4.b** Syntax diagram for *AssignmentStatement*