Except for file variables (see Chapter 9), assignment is possible to variables of any type. The variable (or the function) and the expression must be *assignment compatible*. All the cases for assignment–compatibility are listed below:

1. The variable and the expression are the same type except if that type is a file type (see Chapter 9) or contains a file type as a component in another structured type.
2. The variable is real type and the expression is integer type.
3. The variable and the expression are the same or subranges (see Chapter 5) of the same ordinal type, and the value of the expression lies within the closed interval specified by the type of the variable. The value of the expression must be a value of the type of the variable
4. The variable and the expression are the same set type (see Chapter 8) or are set types with base types which are the same or subranges of the same ordinal type. Either both types or neither type must be packed.
5. The variable and the expression are string types (see Section 6.B) with the same number of elements.

Examples of assignments:

```
Root1  := Pi*X/Y
Root2  := -Root1
Root3  := (Root1 + Root2) * (1.0 + Y)
Danger := Temp > VaporPoint
Count  := Count + 1
Degree := Degree + 10
SqrPr  := sqr(pr)
Y      := sin(X) + cos(Y)
```

4.B. The Procedure Statement

Another kind of simple statement is the *procedure statement*, which activates the named procedure which is a subprogram specifying another set of actions to be performed on data. So far in this tutorial we have used the procedures `Read`, `Readln`, `Write`, and `Writeln` to perform input and output. Procedure statements are discussed fully in Chapter 11.