

## CHAPTER 3: CONTROL STRUCTURE

A computer program is shaped by its data representation and the statements that determine its flow of control. These define the *structure* of a program. There is no sharp distinction between expression and organization; it is more a question of scope. In the previous chapter we were concerned with the details of expressing each statement well. In this chapter and in the next we will concentrate on matters of style that affect the program as a whole.

The control structures of a language provide the framework of a program. These include decision-making with IF and ELSE; looping with DO and WHILE; statement grouping; and procedures or subroutines and functions. The care with which they are used determines how easy it will be to understand the program in the large — in what order things happen, and what controls what. The transformations we made in Chapter 2, such as removing obviously unnecessary GOTO's and statement labels, are simple examples of the proper use of control flow. In this chapter we will go much further.

The easiest construction is the group of statements — a set of operations that are always done together and in sequence. PL/I provides DO-END and BEGIN-END to delimit groups of statements that belong together. In PL/I, branching around a group of statements with THEN GOTO is a sign of “Fortran-think,” a clue that rearrangement is called for.

```
IF PRICE(J) > LOT THEN GO TO X;
/* REDEFINE LOT IF LOWER PRICE IS FOUND */
LOT = PRICE(J);
/* STORE LOCATION OF THE LOWEST PRICE */
LOCATION = J;
X: ...
```

Since PL/I's DO-END permits a group of statements to follow an IF, there is never any need to branch around them. And so there is never any need to invent a label, nor to try and figure out where a GOTO is going, nor to wonder how many ways one can get to a label. Turning “greater than” into “less than or equal to” here lets us introduce a DO-END and eliminate the label and the GOTO. At the same time, we indent to emphasize that the IF controls the two indented lines, and omit the repetitive comments, which obscure the code without conveying information.