```
IF CLOSE_BALANCE < 0 THEN SERV_CHARGE = 7.00;
    ELSE BEGIN;
    IF CLOSE_BALANCE < 100.00 THEN SERV_CHARGE = 2.00;
       ELSE BEGIN;
       IF LOW_BALANCE < 100.00 THEN SERV_CHARGE = 1.50;
          ELSE BEGIN;
          AVE_BALANCE = OPEN_BALANCE + HIGH_BALANCE +
                        LOW_BALANCE + CLOSE_BALANCE;
          IF AVE_BALANCE < 800 THEN SERV_CHARGE = 1.00;
             ELSE BEGIN;
             IF AVE_BALANCE < 1600 THEN SERV_CHARGE = 0.50;
             ELSE SERV_CHARGE = 0; END;
          END;
       END;
    END;
```

The indentation, though clearly systematic, is not a help. Nor do all the extraneous BEGIN-END pairs contribute much. If we eliminate the unnecessary grouping, and indent to show that the program is basically two CASE statements, things clarify remarkably.

```
IF CLOSE_BALANCE < 0 THEN
   SERV_CHARGE = 7.00;
ELSE IF CLOSE_BALANCE < 100.00 THEN
   SERV_CHARGE = 2.00;
ELSE IF LOW_BALANCE < 100.00 THEN
   SERV_CHARGE = 1.50;
ELSE DO;
   AVE_BALANCE = OPEN_BALANCE + HIGH_BALANCE +
      LOW_BALANCE + CLOSE_BALANCE;
   IF AVE_BALANCE < 800 THEN
      SERV_CHARGE = 1.00;
   ELSE IF AVE_BALANCE < 1600 THEN
      SERV_CHARGE = 0.50;
   ELSE
      SERV_CHARGE = 0;
END;
```

A CASE should *not* have each level of ELSE indented, as is often recommended. Placing all the ELSE-IF's of a CASE at one level makes the multi-way nature more clear, and also helps to keep long ones from disappearing off the right side of the page.

Indent to show the logical structure of a program.

Another example of how an ill-chosen layout can hinder comprehension is