You're changing all the `if` statements into a `switch-case` thing. Be careful what you type. When you're done, double-check Lines 19 through 37 in the source code to make sure that you got it right. (The first few lines of the program don't change.)

Compile. Fix any errors or typos. Note that those are colons — not semicolons — on the `case` lines. The character constants are enclosed in single quotes. The word *default* also ends in a colon.

Run. You should see no difference in the output. Internally, however, you have converted an ugly string of `if-else` statements into an elegant decision-making structure: the `switch-case` loop (or "structure thing").

- Detailed information on what happens in a `switch-case` loop is covered in the next section.
- The `switch` command in Line 19 takes the single character typed at the keyboard (from Line 18) and tells the various `case` statements in its curly braces to find a match.
- Each of the `case` statements (Lines 21, 24, 27, and 30) compares its character constant with the value of variable `c`. If there's a match, the statements belonging to that `case` are executed.
- The `break` in each `case` statement tells the compiler that the `switch-case` thing is done and to skip the rest of the statements.

- If the `break` is missing, execution falls through to the next group of `case` statements, which are then executed, no matter what. Watch out for that! Missing `break`s are the bane of `switch-case` loops.
- The final item in the `switch-case` thing is `default`. It's the option that gets executed if no match occurs.

The Old `switch-case` Trick

This is one booger of a command to try to become comfy with. Although `switch-case` things are important, they contain lots of programming finesse that many beginners stumble over. My advice is to work through the programs in this chapter to get a feel for things and then check mark bullets in this section for review purposes or to figure out what went wrong when things don't work.

The `switch` keyword is used to give your programs an easy way to make multiple-choice guesses. It should replace any long repetition of `if-else` statements you're using to make a series of comparisons.