The Helpful RULES Program

I just can't let you go without a program in this chapter. To help you understand the most easily offended C rules, I have summarized them in the following program. It displays several lines of text that remind you of the basic rules of C that you know about:

```
#include <stdio.h>

int main()
{
 printf("Braces come in pairs!");
 printf("Comments come in pairs!");
 printf("All statements end with a semicolon!");
 printf("Spaces are optional!");
 printf("Must have a main function!");
 printf("C is done mostly in lowercase.\
    It's a case-sensitive language.");
 return(0);
}
```

Type the preceding source code into your editor. Save the code to disk as RULES.C. Compile and run.

The resulting program is named RULES, and you can run it whenever you need a reminder about some basic C do's and don'ts.

- This program is really no different from those shown in previous chapters. It merely has more `printf()` functions.
- The final `printf()` function (in Line 10) may seem a little odd. That's because it's split between two lines. This weird contraption is covered later in this chapter.

The importance of being \n

Did you notice something odd about the output of the RULES program? Yes, it resembles an ugly clot of text:

```
Braces come in pairs!Comments come in pairs!All statements
           end with a semicolon!Spaces are optional!Must have
           a main function!C is done mostly in lowercase.
           It's a case-sensitive language.
```

The source code looks okay, but what's missing from the output is the character you get when you press the Enter key, or what's called the *newline* character. You have seen it before. It's that weird `\n` thing: