```
    printf("Not once, or twice, but three times a day!\n");
    jerk();
    printf("He insulted my wife, my cat, my mother\n");
    printf("He irritates and grates, like no other!\n");
    jerk();
    printf("He chuckles it off, his big belly a-heavin'\n");
    printf("But he won't be laughing when I get even!\n");
    jerk();
    return(0);
}

/* This is the jerk() function */

void jerk()
{
    printf("Bill is a jerk\n");
}
```

When you're done, resave BIGJERK2.C to disk. Recompile, and you shan't be bothered by the various warning errors again.

- The prototype is basically a rehash of a function that appears later in the program.
- The prototype must shout out what type of function the program is and describe what kind of stuff should be between the parentheses.
- The prototype must also end with a semicolon. This is *muy importanto.*
- I usually copy the first line of the function to the top of the program, paste it in there, and then add a semicolon. For example, in BIGJERK2.C, I copied Line 21 (the start of the `jerk` function) to the top of the source code and pasted it in, adding the necessary `void`s and semicolon.
- No, the `main()` function doesn't have to be prototyped. The compiler is expecting it and knows all about it. (Well, almost. . . .)
- Required prototyping is something they added to the C language after it was first introduced. You may encounter older C source code files that seem to lack any prototyping. Back in the days when such programs were written (before about 1990), this was a common way of doing things.

A sneaky way to avoid prototyping problems

Only the coolest of the C language gurus do this trick — so don't tell anyone that you found out about it in a *For Dummies* book! Shhhh!