

5. Remember to send the proper values when you're calling the function.

Because the function is required to eat values, you must send them along. No more empty parentheses! You must fill them, and fill them with the proper type of value: integer, character, floater — whatever. Only by doing that can the function properly do its thing.



- ✓ The parameter is referred to as an *argument*. This term gives you a tiny taste of C's combative nature.
- ✓ The name you give the function's parameter (its passed-along variable, argument, or whatever) is used when you're defining and prototyping the function, and inside the function.
- ✓ You can treat the function's parameter as a local variable. Yeah, it's defined in the prototype. Yeah, it appears on the first line. But, inside the function, it's just a local variable.
- ✓ By the way, the variable name used inside the function must match the variable name defined inside the function's parentheses. More on this later.
- ✓ Information on passing strings to functions is provided in my book *C All-in-One Desk Reference For Dummies* (Wiley).
- ✓ Sending a value to a function or getting a value back isn't the same as using a global variable. Although you can use global variables with a function, the values the function produces or generates don't have to be global variables. (Refer to Chapter 21 for more information about global variables.)



An example (and it's about time!)

Blindly type the following program, a modification of the BIGJERK.C cycle of programs you work with in Chapter 20:

```
#include <stdio.h>

void jerk(int repeat);

int main()
{
    printf("He calls me on the phone with nothing say\n");
    printf("Not once, or twice, but three times a day!\n");
    jerk(1);
    printf("He insulted my wife, my cat, my mother\n");
    printf("He irritates and grates, like no other!\n");
    jerk(2);
    printf("He chuckles it off, his big belly a-heavin'\n");
    printf("But he won't be laughing when I get even!\n");
    jerk(3);
}
```