```
void main()
{
    printf("He calls me on the phone with nothing say\n");
•   printf("Not once, or twice, but three times a day!\n");
        {
        jerk();r("Bill is a jerk!\n");

  • printf("He insulted my wife, my cat, my mother\n");
•   printf("He irritates and grates, like no other!\n");
        {
        jerk();r("Bill is a jerk!\n");

  • printf("He chuckles it off, his big belly a-heavin'\n");
•   printf("But he won't be laughing when I get even!\n");
        {
        jerk();r("Bill is a jerk!\n");

}
•
•
```

Figure 20-1:
How a
function
works in a
program.

✔ The computer still reads instructions in the source code from the top
down in the main function. However, when the computer sees another
function, such as jerk(), it temporarily sidesteps to run the instructions
in that function. Then, it returns back to where it was.

✔ Keep in mind that not all functions are as simplistic as jerk(). Most of
them contain many lines of code — stuff that would be too complex and
redundant to use all over the place in a program.

Prototyping Your Functions

Prototyping refers to the art of telling the compiler what's demanded of a func-
tion. It may seem like a silly thing to do, but it is in fact a good way to ensure
that functions are used properly — plus it helps you keep track of your code. I
know, it sounds a little like the compiler doesn't trust you. But you probably
don't trust it much either, so the respect is mutual.

✔ *Proto* comes from the Greek word for *first.*

✔ *Typing* comes from the Latin word for "what you do on a keyboard."