The getchar() function

Fortunately, you're not stuck with scanf() for reading in individual keys from the keyboard. The C language also has a function named getchar(), which is used specifically to read a single character from the keyboard. Here's the format:

```
var=getchar();
```

var is a character variable. It holds whatever character is typed at the keyboard. *var* is followed by an equal sign and then getchar and two parentheses hugging nothing. This function is a complete statement and ends with a semicolon.

The getchar() function causes your program to pause and wait for a key to be typed at the keyboard. getchar() sits and waits. Sits and waits. Sit. Wait. Sit. Wait. When a key is typed and then Enter is pressed, that character's "value" slides across the equal sign and is stored in the character variable.

The following is the update to the FAVKEY1.C program, this time replacing the sordid scanf() function with the better getchar() function:

```
#include <stdio.h>

int main()
{
    char key;

    puts("Type your favorite keyboard character:");
    key=getchar();
    printf("Your favorite character is %c!\n",key);
    return(0);
}
```

Edit the source code for FAVKEY1.C, changing only Line 8 and replacing the scanf() function with getchar(), as just shown. Save the new file to disk as FAVKEY2.C. Compile and run!

The output is the same as for the first version of the program; and you still have to press the Enter key to enter your favorite key value.

- Yes, it seems silly that you have to type Enter when entering a single character. That's just the way getchar() works. (And I hate it.)
- There are ways to read the keyboard more interactively, where pressing the Enter key isn't required. I cover these methods in my book *C All-in-One Desk Reference For Dummies* (Wiley).