Like `scanf()` reading in text, `gets()` requires a `char` variable to store what's entered. It reads everything typed at the keyboard until the Enter key is pressed. Here's the format:

```
gets(var);
```

`gets()`, like all functions, is followed by a set of parentheses. Because `gets()` is a complete statement, it always ends in a semicolon. Inside the parentheses is *var,* the name of the string variable text in which it is stored.

Another completely rude program example

The following is the INSULT1.C program. This program is almost identical to the WHORU.C program, introduced in Chapter 4, except that `gets()` is used rather than `scanf()`.

```
#include <stdio.h>

int main()
{
    char jerk[20];

    printf("Name some jerk you know:");
    gets(jerk);
    printf("Yeah, I think %s is a jerk, too.\n",jerk);
    return(0);
}
```

Enter this source code into your editor. Save the file to disk and name it INSULT1.C.

Compile the program. Reedit the text if you find any errors. Remember your semicolons and watch how the double quotes are used in the `printf()` functions.

Run the resulting program. The output looks something like this:

```
Name some jerk you know:Bill
Yeah, I think Bill is a jerk, too.
```

- `gets()` reads a variable just like `scanf()` does. Yet no matter what reads it, the `printf()` statement can display it.
- `gets(var)` is the same as `scanf("%s",var)`.
- If you get a warning error when compiling, see the next section.