



- ✓ Having a `for` loop inside a `while` loop is referred to as a *nested loop*. Note that both loops don't need to be of the same type (two `for` loops or two `while` loops).
- ✓ A nested loop is basically one loop spinning 'round inside another loop.
- ✓ The first loop, or outside loop, ticks off first. Then, the inside loop ticks off, looping as many times as it does. After that, the outside loop ticks off another one, and then the inside loop is repeated entirely again. That's how they work.
- ✓ Keep separate the variables associated with one loop or another. For example, the following two `for` loops are nested improperly:

```
for(x=0;x<5;x++)
    for(x=5;x>0;x--);
```

Because `x` is used in both loops, these nested loops don't behave as you expect. This loop is infinite, in fact, because both are manipulating the same variable in different directions.



- ✓ This disaster probably isn't apparent to you. You write some huge program and nest two `for` loops miles apart without thinking about it, by using your favorite variable `x` (or `i`) in each one. Those kind of bugs can wreck your day.



- ✓ The way to avoid messing up nested loops is to use different variables with each one — for example, `a` or `b`, or `i1` and `i2`, or even something descriptive, such as `start` and `delay`, as used in the `COUNTDOWN.C` example.
- ✓ That nested `for` loop in `COUNTDOWN.C` ends with a semicolon, indicating that it doesn't "own" any statements that are repeated. Here's another way you could format it:

```
for(delay=0;delay<100000;delay++)
    ;
```

This example shows you that the `for` loop doesn't have any statements worth repeating. It just sits and spins the microprocessor, wasting time (which is what you want).

- ✓ Although `delay` loops, such as the one shown in `COUNTDOWN.C`, are common, a better way exists. That is to use the computer's internal clock to time a delay of a specific duration. I show you an example in *C All-in-One Desk Reference For Dummies* (Wiley).
- ✓ My first IBM PC — some 20 years ago — required a `delay` loop that counted to only 10,000 for about a half-second pause between each line displayed. Today's computers are much, much faster — obviously!

