Tip 5: Initialize wherever the first use may be

Ben Klemens

9 October 2011

level: still pretty basic
purpose: not think about declarations so much

I see code like this pretty often:

```
int main(){
    char *head;
    int i;
    double ratio, denom;

    denom=7;
    head = "There is a cycle to things divided by seven.";
    printf("%s\n", head);
    for (i=0; i< 10; i++){
        ratio = i/denom;
        printf("%g\n", ratio);
    }
}
```

We have three or four lines of introductory material (I'll let you decide how to count the white space), followed by the routine.

This is somewhat a matter of style, but I think this looks archaic, and I've heard from a few folks who learned to code via untyped scripting languages for whom the introductory declarations are a direct and immediate turn-off. Variables still have to have a declared type, but here's how I'd write the code to minimize the burden:

```
int main(){
    double denom=7;
    char *head = "There is a cycle to things divided by seven.";
    printf("%s\n", head);
    for (int i=1; i<= 6; i++){
        double ratio = i/denom;
        printf("%g\n", ratio);
    }
}
```

Here, the declarations happen as needed, so the onus of declaration reduces to sticking a type name before the first use. If you have color syntax highlighting, then the declarations are still easy to spot (and if you don't have color, golly, get a text editor that supports it—there are dozens to hundreds to choose from!).

Also, by the rule that you should keep the scope of a variable as small as possible, we're pushing the active variable count on earlier lines that much lower. When you have a too-long function running a page or two, this can start to matter. As for the index, it's a disposable convenience for the loop, so it's natural to reduce its scope to exactly the scope of the loop.

Did you notice how I've been specifying `-std=gnu99` in the sample makefiles and other calls to `gcc`? Declaring the iterator variable for the `for` loop wasn't valid in the mid-1990s, and GCC is still stuck in using that as the norm. It's annoying, I know, but that's a minor kvetch about a program as awesome as `gcc`. If you're not already compiling via an alias or a makefile, you might want to add `alias gcc="gcc -std=gnu99"` to your `.bashrc` to get GCC to always use the 1999 standard.