

## Tip 40: Initialize arrays and structs with zeros

Ben Klemens

22 December 2011

**level:** `struct user`

**purpose:** don't leave values undefined

If you declare a variable inside a function, then C won't zero it out automatically. [Which is odd, because these are known as automatic variables. For more on the memory types, see Tip #20 (Entry #070).] I'm guessing that the rationale here is a speed savings: when setting up the frame for a function, zeroing out bits is extra time spent, which could potentially add up if you call the function a million times and it's 1985.

But here in the present, leaving a variable undefined is asking for trouble.

For simple numeric data, set it to zero on the line where you declare the variable. For pointers, including strings, set it to `NULL`. OK, that's easy enough, as long as you remember.

For structs and arrays of constant size, recall how last time (Entry #090) I showed you that if you use designated initializers but leave some elements blank, those blank elements get set to zero; you can set the whole structure to zero by assigning a complete blank. Here's a do-nothing program to demonstrate the idea:

```
typedef struct {
    int la, de, da;
} ladedata_t;

int main(){
    ladedata_t emptystruct = {};
    int ll[20] = {};
}
```

Isn't that easy and sweet.

Now for the sad part: let us say that you have a variable-length array (i.e., one whose length is set by a run-time variable). The only way to zero it out is via `memset`:

```
int main(){
    int length=20;
    int ll[length];
    memset(ll, 0, 20*sizeof(int));
}
```

This is bitter in exact proportion to the sweetness of initializing fixed-length arrays.  
[You can blame ISO C standard part 6.7.8§3 for this, because it insists that variable length arrays can't be initialized. I say the compiler should be able to work it out....] So it goes.

**To do:**

Write yourself a macro to declare a variable-length array and set all of its elements to zero. You'll need inputs listing the type, name, and size.