

Plain English Programming — Nested IFs

[gerryrzeppa](#)[Uncategorized](#)

May 6, 2018

3 Minutes

This is a picture of Sharon, the Mother of all Osmosians:



Her favorite saying is, “Just tell us what you need, sonny, and we’ll show you how to live without it.” She insisted, in fact, that we use nothing but black dots of different sizes to show you what she looks like.

Another of her favorite sayings is about nested IFs. She says, “Two deep is too deep, boys. No nesting.”

Now everyone knows that “If Momma ain’t happy, ain’t nobody happy,” so when we developed our Plain English compiler we intentionally left out nested IFs. Turns out she was right, as usual — they’re not necessary. Search through the 25,000 Plain English sentences that comprise our system — unique desktop, simplified file manager, elegant text editor, handy hexadecimal dumper, native-code-generating compiler/linker, and WYSIWYG document editor — and you won’t find a single nested IF. Nada. Zip. Zilch. No “elses” either.

An Example

So how does one live without nested IFs? Let’s consider an example given by R S Abhishek, a Research Associate at Ati Motors that popped up when I asked Google for a “real life nested IF statement”:

```

if (age < 60){
    if (income <= 250000){
        tax_percent = 0
    } else if (income >= 250001 && income <= 500000){
        tax_percent = 0.1
    } else if (income >= 500001 && income <= 1000000){
        tax_percent = 0.2
    } else {
        tax_percent = 0.3
    }
} else if (age >= 60 && age < 80){
    if (income <= 300000){
        tax_percent = 0
    } else if (income >= 300001 && income <= 500000){
        tax_percent = 0.1
    } else if (income >= 500001 && income <= 1000000){
        tax_percent = 0.2
    } else {
        tax_percent = 0.3
    }
} else {
    if (income <= 500000){
        tax_percent = 0
    } else if (income >= 500001 && income <= 1000000){
        tax_percent = 0.2
    } else {
        tax_percent = 0.3
    }
}

```

Twenty-seven lines there, with lots of indentation, highlighted keywords, and too many squiggly braces for me to count. Now Let's consider the non-nested, Plain English equivalent:

To get a tax rate for an age and an income:

If the age is less than 60, get the tax rate for young people with the income; exit.
 If the age is between 60 and 79, get the tax rate for old folks with the income; exit.
 If the income is less than or equal to 500000, put 0 into the tax rate, exit.
 If the income is between 500001 and 500000, put 2/1000 into the tax rate; exit.
 Put 3/1000 into the tax rate.

To get a tax rate for young people given an income:

If the income is less than or equal to 250000, put 0 into the tax rate, exit.
 If the income is between 250001 and 500000, put 1/1000 into the tax rate; exit.
 If the income is between 500001 and 1000000, put 2/1000 into the tax rate; exit.
 Put 3/1000 into the tax rate.

To get a tax rate for old folks given an income:

If the income is less than or equal to 300000, put 0 into the tax rate, exit.
 If the income is between 300001 and 500000, put 1/1000 into the tax rate; exit.

If the income is between 500001 and 1000000, put 2/1000 into the tax rate; exit.
Put 3/1000 into the tax rate.

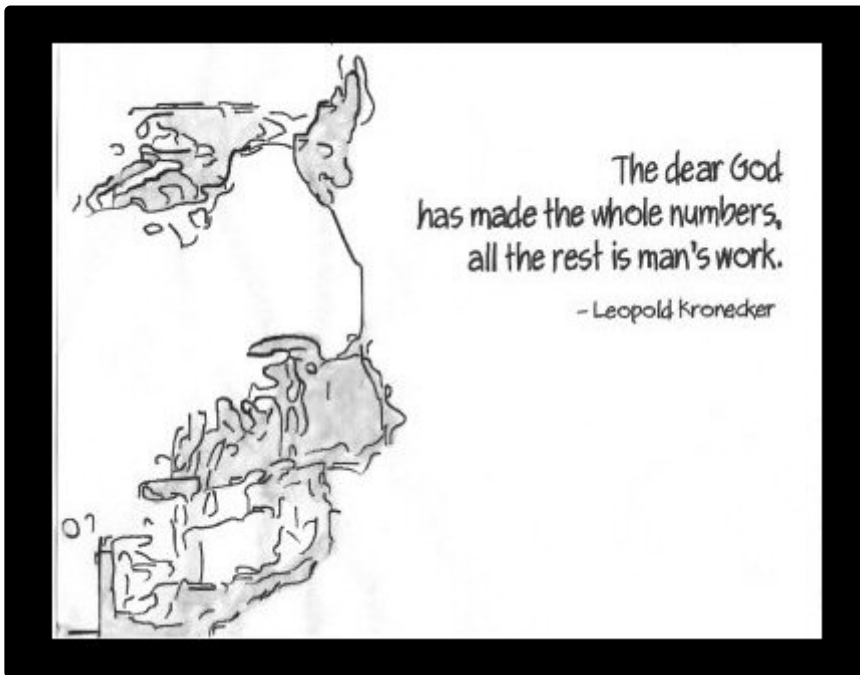
Just eighteen lines (including the optional blanks between the routines), and a mere handful of punctuation marks — all used in the usual ways.

The trick, you see, is to properly factor the problem by handling (and thus eliminating) special cases at the top, exiting after each. The normal/default/fall-through case then appears, unconditioned, at the bottom.

Some notes

If you're wondering whether "between" in Plain English is inclusive or exclusive, just do what we did. Head down to the local Walmart and ask the folks coming and going to "Pick a number between 1 and 10." You'll find that lots of people choose 1 and 10, which shows that the *normal* interpretation of "between" is inclusive.

And if you're wondering why we put "3/1000" into the tax rate (instead of 0.3, like the other guy), it's because Plain English is a whole-number only language. Why? Because our Osmosian Mom used to put us to sleep with stories from an ancient Osmosian adventure book entitled, *The Epic Battle for the Soul of Mathematics: Kronecker vs Cantor*. I remember, in fact, this painting hanging over my bed:



But that's really a whole 'nother story. It will have to wait for another article. Suffice it to say, at this juncture, that Mom was right about nested IFs — you don't need 'em.