

Re: Code consolidation

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-08/msg03111.html

- *From:* Chris McDonald <chris@xxxxxxxxxxxxxxxx>
 - *Date:* Wed, 31 Aug 2005 03:09:32 +0000 (UTC)
-

Aaron Jackson <none@xxxxxxxx> writes:

>I am in the process of writing my first large scale program and I find
>myself in the position of writing several copies of functions that do
>the same thing, with mostly the same code, but only slightly different.
>For example:

```
>double  
>Chisqrd1(double *params, double *weight) { // CHANGE  
> int i;  
> double deltax, chiSqr;  
> xy_t *data;  
  
> chiSqr = 0.0;  
> data = dataList;  
> for (i = 0; data != NULL; i++) {  
> deltax = data->y - netCarriers(params, data->x); // CHANGE  
> chiSqr += weight[i] * deltax * deltax;  
> data = data->next;  
> }  
> return(chiSqr);  
>}
```

```
>double  
>Chisqrd2(dopant_t *dopants, double *weight, double *Ef, double Eg) { //  
>CHANGE  
> int i;  
> double deltax, chiSqr;  
> xy_t *data;  
  
> chiSqr = 0.0;  
> data = dataList;  
> for (i = 0; data != NULL; i++) {  
> deltax = data->y - netIonized(dopants, data->x, Ef[i]); // CHANGE  
> chiSqr += weight[i] * deltax * deltax;  
> data = data->next;  
> }  
> return(chiSqr);  
>}
```

Re: Code consolidation

>I was wondering if there are common ways to cut down on this type of
>duplication without adding too much to the overall complexity. Does
>anybody have any suggestions? Thanks.

I suggest making all of your functions, netXXXX, to be of the same
type signature (they all receive the same parameter types (even if they
don't use them all), and all return the same type), and then passing the
required netXXXX function as a parameter to a single Chisqrd function.

—

Chris.

.

• **References:**

◆ **Code consolidation**

◇ *From:* Aaron Jackson

- Prev by Date: **Re: C FAQ wiki**
- Next by Date: **Re: Why C/C++ errors are SO obscure/devious??**
- Previous by thread: **Code consolidation**
- Next by thread: **Re: Code consolidation**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**