The following is suggested as a response to DR 413 on designated initializers.

**Committee Discussion**

6.7.9 paragraphs 17-18 specify that each designator list affects only the smallest subobject to which the designator list refers. As a result, the second clause of paragraph 19 occurs once for the greater object as a whole, filling in only those parts of the whole object that were never initialized explicitly.

**Technical Corrigendum**

Add a new footnote at the end of 6.7.9 paragraph 18.

newfootnote) The subobject to be initialized is the smallest subobject that is specified by the designator list.

Replace 6.7.9 paragraph 19 with the following.

oldfootnote) The initialization shall occur in initializer list order, each initializer provided for a particular subobject overriding any previously listed initializer for the same subobject.

Add the following to the end of paragraph 21.

Implicit initialization does not override explicit initialization.

Add the following example to 6.7.9.

```c
typedef struct {
    int k;
    int l;
    int a[2];
} T;

typedef struct {
    int i;
    T t;
} S;
```
\[
T \ x = \{ .l = 43, .k = 42, .a[1] = 19, .a[0] = 18 \};
\]

\[
void f(void)
\{
    S \ l = \{ 1, .t = x, .t.l = 41, .t.a[1] = 17\};
\}
\]

The value of \( l.t.k \) is 42, because implicit initialization does not override explicit initialization.