Subject: Re: Order of member initialization
Posted by peterh on Wed, 21 Sep 2022 09:49:29 GMT
View Forum Message <> Reply to Message

Yes, I did not stop testing after I wrote this and discovered problems.

Did you see my previous example some messages above, where the array is allocated from Heap?

```
class B : public A{
public:
 Array<int> &b_array;
 B() : A(*(new Array<int>)),b_array(A::array){
        .....
        }
```

This works securely, I believe, because the array is constructed and initialized to an empty array when it is allocated.
This ensures the array is constructed, before A and B are constructed.

I do however not know if C++ guarantees that A is constructed before B. This could be different with different compilers.
I do not find much about this in my books and online, and this could mean it is implementation and compiler vendor dependant.
However, if this can be guaranteed, then this should be a clean solution.

The most clean way is probably: create an empty array and give it to the B::B(....) constructor as an argument. Then B has no need to create and initialize the array.