
Subject: Re: Order of member initialization

Posted by [jjacksonRIAB](#) on Tue, 20 Sep 2022 15:06:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

I found out the same thing a while ago and then it made sense to me. The base class is initialized first, then the initializers. I think what I ended up doing to get around that, and I still don't know if it's correct, is that I would so something like create a context, a struct holding the Array as well as a function that would return *this, I'd inherit from it instead and then then I'd pass the ref function to the constructor of the second base class which would then reference it.

Looked something like this:

```
struct BaseA {
    Array<int> a;
    auto& BaseARef() { return *this; }
};

struct BaseB {
    BaseA& baseA;

    BaseB(BaseA& baseA) : baseA(baseA) {
        // do something with baseA.a;
    }
};

struct Whatever : BaseA, BaseB {
    Whatever() : BaseA(), BaseB(BaseARef()) {}
};
```

It was weird and I'm not sure if it's sound - but it did work, and if anyone knows the compiler better, feel free to let me know.
