Subject: Re: Order of member initialization
Posted by jjacksonRIAB on Wed, 21 Sep 2022 17:00:08 GMT
View Forum Message <> Reply to Message

peterh Quote:
I find this answer from stackoverflow very clear:

https://stackoverflow.com/questions/2517050/c-construction-a nd-initialization-order-guarantees

I hope it is correct.

with inherited classes, that's what I thought too (that it proceeded left to right), but I didn't check.

https://en.cppreference.com/w/cpp/language/eval_order

Looks like they call them sequence points and they proceed in order - here's initialization lists:

Quote:
10) In list-initialization, every value computation and side effect of a given initializer clause is sequenced before every value computation and side effect associated with any initializer clause that follows it in the brace-enclosed comma-separated list of initializers.

I should acquaint myself with this stuff better, a lot of things have been changing and what was formerly Undefined Behavior is being addressed. I just learned by accident a few months ago that in C++17 there is type inference now even for template types so you no longer have to write:

```
Vector<int>    x { 1, 2, 3 };
Vector<String> y { "hello", "world" };
```

you can just write:

```
Vector x { 1, 2, 3 };
Vector y { "hello", "world" };
```

and it will fill in the type for you.