Subject: Re: Order of member initialization
Posted by peterh on Tue, 20 Sep 2022 20:59:16 GMT

View Forum Message <> Reply to Message

Tom1 wrote on Tue, 20 September 2022 11:19Hi,
Is there a way to re-order the initialization of class B so that the array gets initialized before class A?

So far I have read, the order of initialization is partially implementation dependent and not defined in C++.
There are some rules, but I do not know exactly yet.

I believe the problem is: The array is not constructed at all because it is never accessed within the scope, only its address is taken.
When it is constructed without arguments, then it should be initialized as an empty array.

When the array is initialized, it is constructed.

This seems to work:

```
#include <Core/Core.h>

using namespace Upp;

class A{
public:
 Array<int> &array;
 A(Array<int> &array_) : array(array_){
  Cout() << "constructing A\n";
  Cout() << "Array Initial item count in A = " << array.GetCount() << "\n";
 }
};

class B : public A{
public:
 Array<int> array;
 B() : A(array), array({}) { //<-------Initialize here
  Cout() << "constructing B\n";
  Cout() << "Array Initial item count in B = " << array.GetCount() << "\n";
 }
};

CONSOLE_APP_MAIN{
 B b;
	Sleep(1000000);
}
```