

# Rule of DesDeMovA

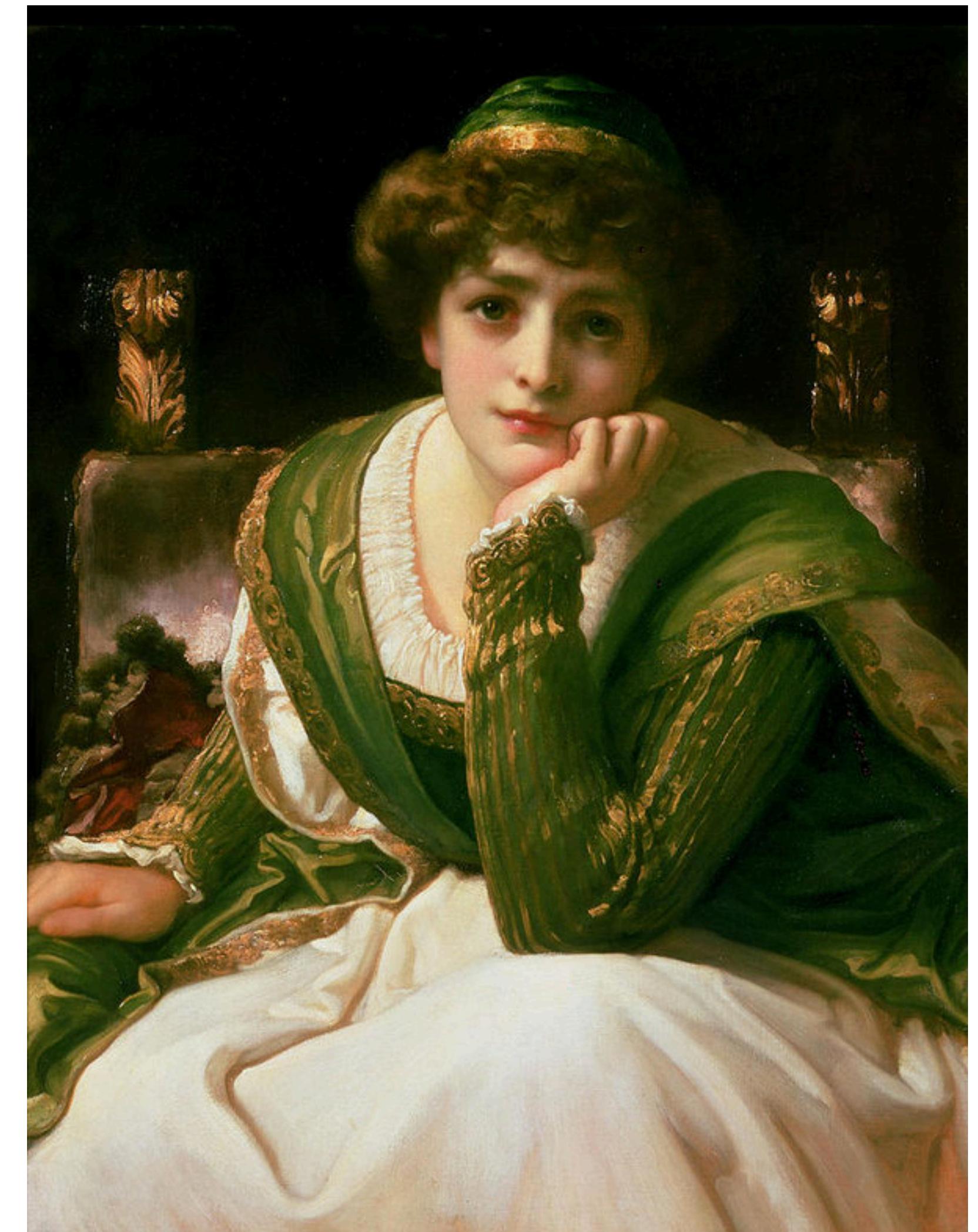


Prof. Peter Sommerlad  
Director of IFS  
C++Now 2019

@PeterSommerlad   
peter.cpp@sommerlad.ch

- **Wikipedia says:**

- Desdemona (/dɛzdə'moʊnə/) is a character in William Shakespeare's play Othello (c. 1601–1604).
- The name derives from Greek δυσ + δαίμων, which means "ill-fated, unfortunate"
- When deciding on which and how to implement C++ class special member functions, many classes become "ill-fated, unfortunate"
- Do not be like Desdemona, use DesDeMovA !



# Do you Remember: What Special Member Functions Do You Get?

What you get

	<b>default constructor</b>	<b>destructor</b>	<b>copy constructor</b>	<b>copy assignment</b>	<b>move constructor</b>	<b>move assignment</b>
<b>What you write</b>	<b>nothing</b>	defaulted	defaulted	defaulted	defaulted	defaulted
	<b>any constructor</b>	not declared	defaulted	defaulted	defaulted	defaulted
	<b>default constructor</b>	<u>user declared</u>	defaulted	defaulted	defaulted	defaulted
	<b>destructor</b>	defaulted	<u>user declared</u>	defaulted (!)	defaulted (!)	not declared
	<b>copy constructor</b>	not declared	defaulted	<u>user declared</u>	defaulted (!)	not declared
	<b>copy assignment</b>	defaulted	defaulted	defaulted (!)	<u>user declared</u>	not declared
	<b>move constructor</b>	not declared	defaulted	deleted	deleted	<u>user declared</u>
	<b>move assignment</b>	defaulted	defaulted	deleted	deleted	<u>user declared</u>

Howard Hinnant's Table: [https://accu.org/content/conf2014/Howard\\_Hinnant\\_Accu\\_2014.pdf](https://accu.org/content/conf2014/Howard_Hinnant_Accu_2014.pdf)

Note: Getting the defaulted special members denoted with a (!) is a bug in the standard.

## What you get

	<b>default constructor</b>	<b>destructor</b>	<b>copy constructor</b>	<b>copy assignment</b>	<b>move constructor</b>	<b>move assignment</b>
<b>What you write</b>	<b>nothing</b>	defaulted	defaulted	defaulted	defaulted	defaulted
<b>any constructor</b>	not declared	defaulted	defaulted	defaulted	defaulted	defaulted
<b>default constructor</b>	<u>user declared</u>	defaulted	defaulted	defaulted	defaulted	defaulted
<b>destructor</b>	defaulted	<u>user declared</u>	defaulted (!)	defaulted (!)	not declared	not declared
<b>copy constructor</b>	not declared	defaulted	<u>user declared</u>	defaulted (!)	not declared	not declared
<b>copy assignment</b>	defaulted	defaulted	defaulted (!)	<u>user declared</u>	not declared	not declared
<b>move constructor</b>	not declared	defaulted	deleted	deleted	<u>user declared</u>	not declared
<b>move assignment</b>	defaulted	defaulted	deleted	deleted	not declared	<u>user declared</u>

Howard Hinnant's Table: [https://accu.org/content/conf2014/Howard\\_Hinnant\\_Accu\\_2014.pdf](https://accu.org/content/conf2014/Howard_Hinnant_Accu_2014.pdf)

Note: Getting the defaulted special members denoted with a (!) is a bug in the standard.

## What you get

	<b>default constructor</b>	<b>destructor</b>	<b>copy constructor</b>	<b>copy assignment</b>	<b>move constructor</b>	<b>move assignment</b>
<b>nothing</b>	defaulted	defaulted	defaulted	defaulted	defaulted	defaulted
<b>any constructor</b>	not declared	defaulted	defaulted	defaulted	defaulted	defaulted
<b>default constructor</b>	<u>user declared</u>	defaulted	defaulted	defaulted	defaulted	defaulted
<b>destructor</b>	defaulted	<u>user declared</u>	defaulted (!)	defaulted (!)	not declared	not declared
<b>copy constructor</b>	not declared	defaulted	<u>user declared</u>	defaulted (!)	not declared	not declared
<b>copy assignment</b>	defaulted	defaulted	defaulted (!)	<u>user declared</u>	not declared	not declared
<b>move constructor</b>	not declared	defaulted	deleted	deleted	<u>user declared</u>	not declared
<b>move assignment</b>	defaulted	defaulted	deleted	deleted	not declared	<u>user declared</u>

Howard Hinnant's Table: [https://accu.org/content/conf2014/Howard\\_Hinnant\\_Accu\\_2014.pdf](https://accu.org/content/conf2014/Howard_Hinnant_Accu_2014.pdf)

Note: Getting the defaulted special members denoted with a (!) is a bug in the standard.

What you write

	default constructor	copy assignment	move constructor	move assignment
nothing	defaulted	defaulted	defaulted	defaulted
any constructor	not declared	not declared	not declared	not declared
default constructor	<u>user declared</u>	defaulted	defaulted	defaulted
destructor	defaulted	<u>user declared</u>	defaulted (!)	defaulted (!)
copy constructor	not declared	defaulted	<u>user declared</u>	defaulted (!)
copy assignment	defaulted	defaulted	defaulted (!)	<u>user declared</u>
move constructor	not declared	defaulted	deleted	deleted
move assignment	defaulted	<u>user declared</u>	deleted	deleted
				=delete

## DesDeMovA



Rule of if  
**Destructor defined**  
**Deleted**  
**Move Assignment**

default constructor

nothing

any constructor

default constructor

destructor

copy constructor

copy assignment

move constructor

move assignment

copy assignment

nothing

any constructor

default constructor

destructor

copy constructor

copy assignment

move constructor

move assignment

Howard Hinnant's Table: [https://accu.org/content/conf2014/Howard\\_Hinnant\\_Accu\\_2014.pdf](https://accu.org/content/conf2014/Howard_Hinnant_Accu_2014.pdf)

Note: Getting the defaulted special members denoted with a (!) is a bug in the standard.

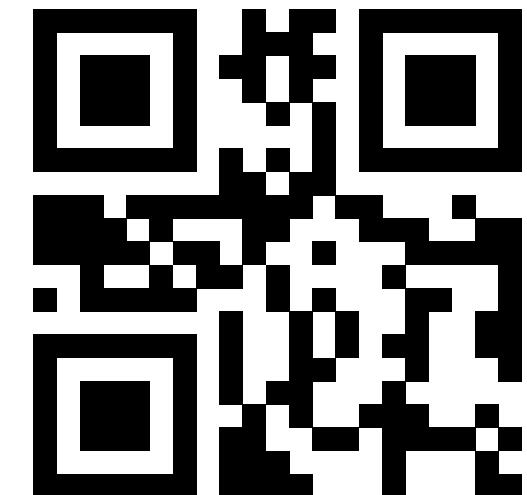


- 1. Rule of Zero**
- 2. Rule of DesDeMovA (no copy, no move for SBRM/RAII and OO-Base classes)**
- 3. Rule of Unique Resource Managers (move-only, no copy)**
- 4. Rule of Five for Resource Managers with Value Semantics, or other really special cases**



**Cevelop**  
Your C++ deserves it

Download IDE at:  
[www.cevelop.com](http://www.cevelop.com)



**Sponsors welcome!**

**Commercial licensing possible!**