

7/13/2016

C++11 Delegating Constructors

dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION



50 Years – Research for
A Life Without Cancer

C++03

```
class ExampleClass
{
public:
    ExampleClass()
        : m_foo(9000),
          m_bar(1)
    {
    }

    explicit ExampleClass(int foo, int bar)
        : m_foo(foo),
          m_bar(bar)
    {
    }

private:
    int m_foo;
    int m_bar;
};
```



C++03

```
class ExampleClass
{
public:
    explicit ExampleClass(int foo = 9000, int bar = 1)
        : m_foo(foo),
          m_bar(bar)
    {
    }

private:
    int m_foo;
    int m_bar;
};
```

C++03

```
class ExampleClass
{
public:
    explicit ExampleClass(int foo = 9000, int bar = 1)
        : m_foo(foo),
          m_bar(bar)
    {
    }

private:
    int m_foo;
    int m_bar;
};
```

Public API is different:

```
ExampleClass() [foo = 9000, bar = 1]
ExampleClass(int foo) [bar = 1]
ExampleClass(int foo, int bar)
```

C++03

```
class ExampleClass
{
public:
    explicit ExampleClass(int foo = 9000, int bar = 1)
        : m_foo(foo),
          m_bar(bar)
    {
    }

private:
    int m_foo;
    int m_bar;
};
```

Public API is different:

```
ExampleClass() [foo = 9000, bar = 1]
ExampleClass(int foo) [bar = 1]
ExampleClass(int foo, int bar)
```

Without “explicit” there would be converting constructors as well:


```
ExampleClass a = 1;
ExampleClass b {1, 2}; // C++11
```

C++11

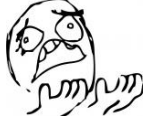
```
class ExampleClass
{
public:
    ExampleClass()
        : ExampleClass(9000, 1) // Delegate
    {
    }

    ExampleClass(int foo, int bar)
        : m_foo(foo),
          m_bar(bar)
    {
    }

private:
    int m_foo;
    int m_bar;
};
```



Sorry for the silly example

- Imagine more complex classes with multiple constructors that do similar stuff in their bodies
 - What you should have done: Common Initialize() function called from these constructors
 - *[Trigger Warning]* What “most” of you did: C&P 
- Now you don't need this extra function anymore
 - Delegate common initialization to most common constructor



Thank you for
your attention!

dkfz.

Further
information
on www.dkfz.de



dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION



50 Years – Research for
A Life Without Cancer