Blow Your Mind

Adrian Kosmaczewski

2008-03-11

Take a careful look at this:

```
#include <iostream>

class Gadget
{
public:
    void sayHello() const
    {
        std::cout << "Gadget!" << std::endl;
    }
};

class Widget
{
public:
    void sayHello() const
    {
        std::cout << "Widget!" << std::endl;
    }
};

template <class T>
class OpNewCreator
{
public:
    T* create()
    {
        std::cout << "Using 'new': ";
        return new T;
    }
};

template <class T>
```

```
class MallocCreator
{
public:
    T* create()
    {
        std::cout << "Using 'malloc': ";
        void* buf = std::malloc(sizeof(T));
        if (!buf) return 0;
        return new(buf) T;
    }
};

template <class T, class B>
class Creator : public T<B>
{
public:
    void exec()
    {
        B* obj = this->create();
        obj->sayHello();
        delete obj;
    }
};

typedef Creator<MallocCreator, Widget> Manager;

int main (int argc, char * const argv[])
{
    Manager obj;
    obj.exec();
    return 0;
}
```

Try changing `MallocCreator` by `OpNewCreator` and `Widget` by `Gadget` in the typedef of line 57, recompile and run; of course you can provide default values, so that

```
template <class T = MallocCreator, class B = Widget>
class Creator : public T<B>
```

so that you just do

```
typedef Creator<>; Manager;
```

I've just started reading "Modern C++ Design" by Andrei Alexandrescu and I've already my head spinning out of orbit. This is amazing (and by giving a quick look at the rest of the book, there's even more incredible stuff there)!