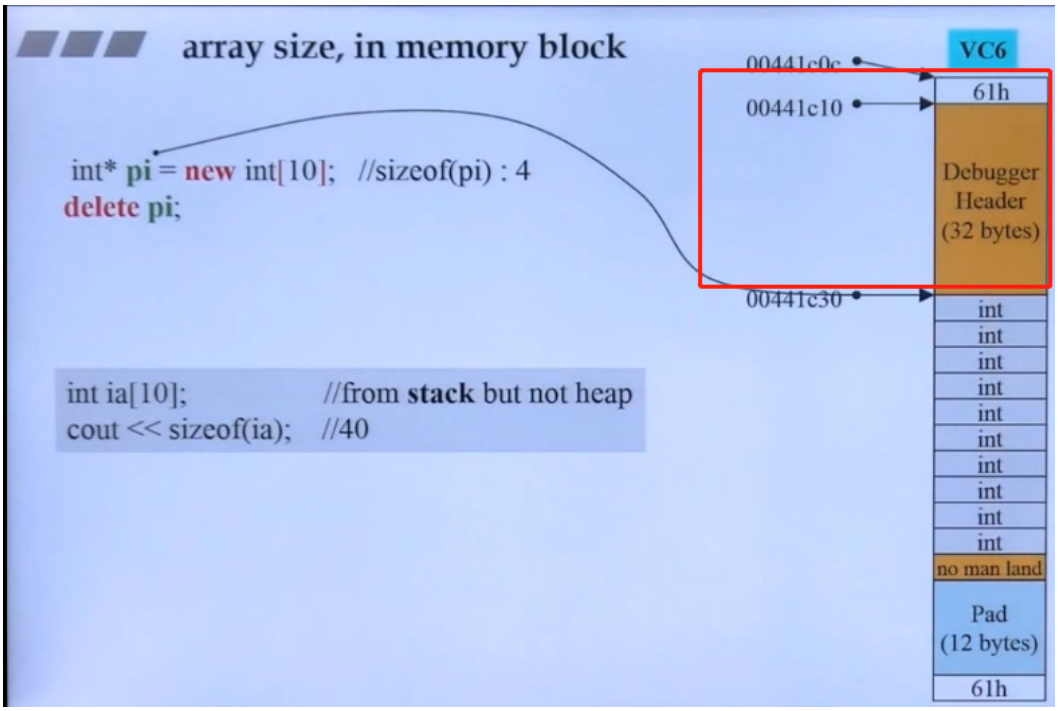
C++内存管理

per-class allocator

引入内存池，一次性分配大内存。因为malloc前面后有个cookie，占用一部分空间。内存池的设计就是处于两个目的：1）速度，减少调用malloc的次数（虽然malloc也挺快的）。 2）空间，减少每个malloc前面的cookie，防止浪费。

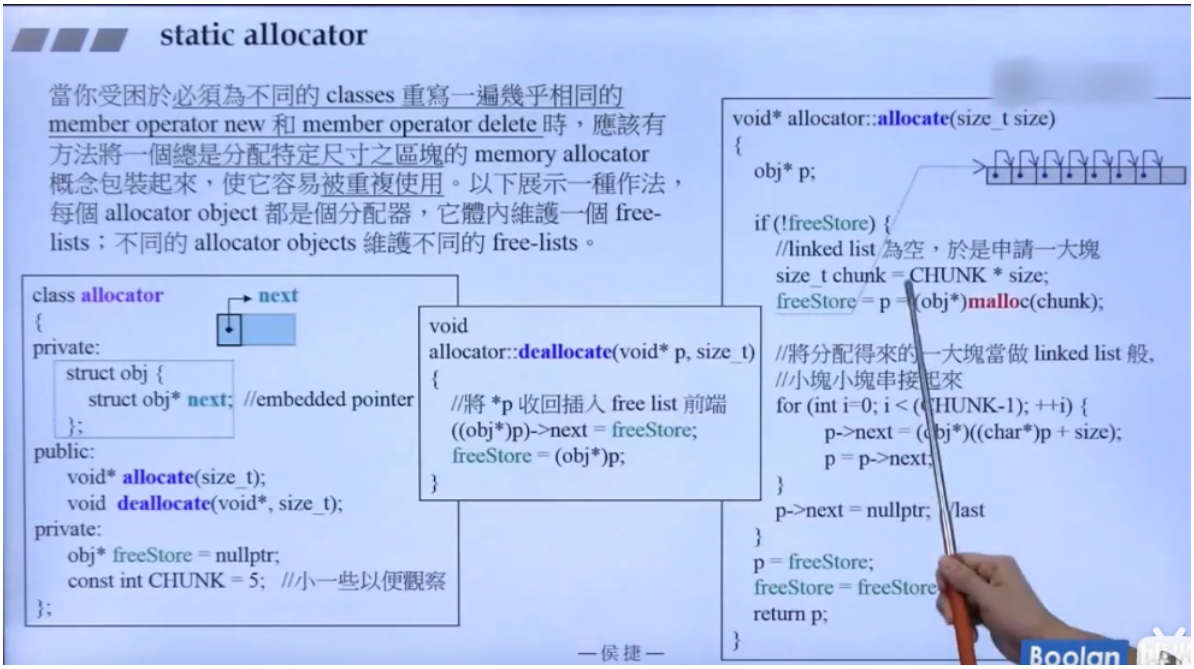


per-class allocator 2

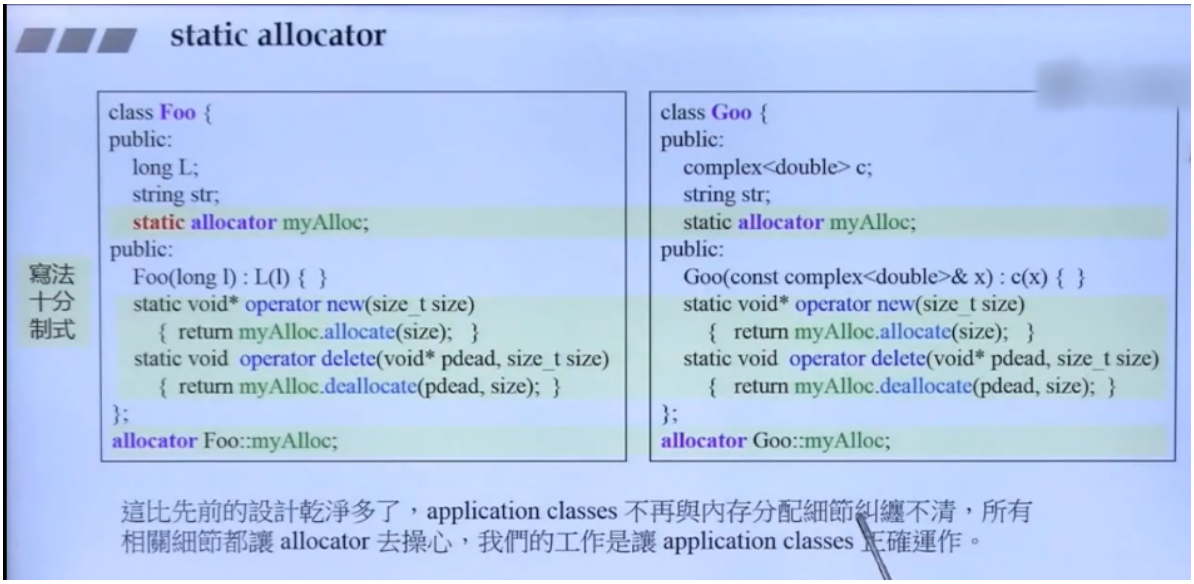
上一节中添加进了一个next指针指向下一个，这样会多出4个字节。在该版本中，使用union，让指针借用前4个字节，避免多分配空间？嵌入式的指针。

static allocator

前边的都是在每个类里单独重载operator new和operator delete，重复性的工作太多。单独设计一个类allocator，里面有allocate和deallocate两个函数，可以被其他类所调用执行。

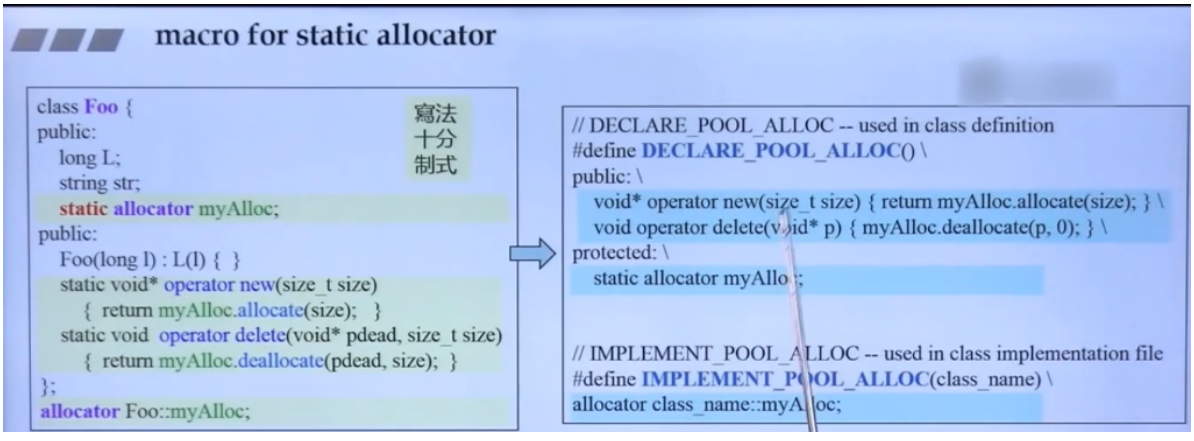


每次只要了5小块，标准的allocator是20？那么这5个之间应该相邻，但是5个和5个之间不一定相邻。



macro for static allocator

设计成了宏



写的时候更加精简了。👇

```
class Foo {
   DECLARE_POOL_ALLOC()
public:
   long L;
   string str;
public:
   Foo(long l) : L(l) { }
};
IMPLEMENT_POOL_ALLOC(Foo)
```

```
class Goo {
   DECLARE_POOL_ALLOC()
public:
   complex<double> c;
   string str;
public:
   Goo(const complex<double>& x) : c(x) { }
};
IMPLEMENT_POOL_ALLOC
```

Boolan

new handler

调用new失败前不止一次调用new handler，这个可以由程序员设定， 相当于在抛异常前由你来决定该怎么办。