

C 中要求（或禁止）对象产生于heap中 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/253/2021\\_2022\\_C\\_\\_\\_E4\\_B8\\_AD\\_E8\\_A6\\_81\\_E6\\_c97\\_253830.htm](https://www.100test.com/kao_ti2020/253/2021_2022_C___E4_B8_AD_E8_A6_81_E6_c97_253830.htm) 要求对象产生于Heap之中

考虑如下代码：`class HeapClass { public: void Destory() const {0delete this.} private: ~HeapClass(){} }. HeapClass* ptr = new HeapClass. ptr->Destory(). 这样的调用真是很厉害，想生成非Heap对象都不成了。对于继承和组合的情况不想多说了，比较无趣的说。判断某个对象是否位于Heap内 考虑如下代码：new HeapClass(* new HeapClass). 你觉得编译器应该怎么做？`

1.调用operator new 2.调用Constructor 3.调用第二个operator new 4.调用第二个Constructor 但是可以让人足够惊讶，编译器对此并不做承诺，所以实际的实现可能是：1.调用operator new 2.调用第二个operator new 3.调用Constructor 4.调用第二

个Constructor 而VC6是这样实现的。`class HeapClass { private: void* operatornew[](size_t size). typedef const void * RawAddress. void operator0delete[](void* ptr). public: void operator0delete(void* ptr) { printf("0delete\n"). ::operator0delete(ptr). m_address.erase(std::remove(m_address.begin(),m_address.end(),ptr),m_address.end()). return. } void* operatornew(size_t size) { printf("new\n"). void * ptr = ::operatornew(size). m_address.push_back(ptr). return ptr. }`

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)