

# 第十三章 动态对象

## 1. 对象的创建

当一个C++的程序被执行，系统会自动干两件事：

①分配内存

②

a. 静态存储区[静态变量、常量]

使用构造函数创建对象

b. 栈[本地变量&对象]

c. 堆[动态变量&对象]

如何创建/销毁动态对象？

-使用 new/new [] 创建

→ 核心是调用了

-使用 delete/delet [] 销毁

→ malloc/free 函数和 构造函数  
析构函数

## 2. new和delete的重载

### 13.2 Overloading new & delete

```
void* operator new(size_t size) {  
    void *t = malloc(size);  
    NUM++;  
    // Here constructor is called if type is CLASS.  
    return t;  
}  
  
void operator delete(void* p) {  
    // Firstly destructor is called if type is CLASS  
    NUM--;  
    if (p != nullptr) {  
        // Avoid using cout because global operator new is called when creating cout.  
        printf("%d, %d\n", *(int*)p, NUM);  
        free(p);  
    }  
}
```

该参数为后一类型的内存大小  
即为size 吋(类型)

全重载

此处不用cout是因为会使用 new 函数  
创建类对象

```
#include <stdio.h>  
#include <malloc.h>  
  
int NUM = 0;  
  
void main()  
{  
    int *a = new int(10);  
    int *b = new int(20);  
    int *c = new int(30);  
  
    delete a;  
    delete b;  
    delete c;  
}
```

一般情况下, 不建议重载 new 和 delete 函数, 除非开着  
希望借助两个函数进行其它更多操作