# 定义对象时new后面加不加括号？

加括号调用没有参数的构造函数，不加括号调用默认构造函数或唯一的构造函数，看需求

C++在new时的初始化的规律可能为：对于有构造函数的类，不论有没有括号，都用构造函数进行初始化；如果没有构造函数，则不加括号的new只分配内存空间，不进行内存的初始化，而加了括号的new会在分配内存的同时进行初始化。

# 定义对象时加不加new？

<https://blog.csdn.net/chang_ge/article/details/52750393>

A a；和A b = new A();

1.前者在堆栈中分配内存，后者为动态内存分配，在一般应用中是没有什么区别的，但动态内存分配会使对象的可控性增强。

2.不加new在栈中分配内存

3.大程序用new，小程序直接申请

4.只是把对象分配在堆栈内存中

5.new必须delete删除，不用new系统会自动回收内存

下面是自己总结的一些关于new创建类对象特点：

new创建类对象需要指针接收，一处初始化，多处使用

new创建类对象使用完需delete销毁

new创建对象直接使用堆空间，而局部不用new定义类对象则使用栈空间

new对象指针用途广泛，比如作为函数返回值、函数参数等

频繁调用场合并不适合new，就像new申请和释放内存一样

# 默认初始化和值初始化：

<https://blog.csdn.net/u010029439/article/details/79519655>

**默认初始化有几种情况：** ① 全局变量，默认初始化为0；② 内建类型的局部非静态变量，默认不被初始化，结果是值未定义；③ 类类型初始化的值，由默认构造函数的定义决定。

**值初始化有几种情况：**① 内建类型的局部静态变量（对应上面的②），值初始化为0； ②使用 Type（）显式地请求一个类对象的初始值时，值初始化。③创建了数组但是没有给数组成员初始值或者初始值个数小于成员个数，按成员类型进行值初始化，string初始化为空字符串，int初始化为0.

其实，只是叫法不一样， 记住各种初始化情况的内容比较重要。 表现上没有很明显的区分， 比如默认初始化的①和值初始化的①，表现就很像。

C++的初始化有很多方式：默认初始化，值初始化，直接初始化，拷贝初始化，列表初始化。这些方式之间有什么区别与联系呢？我们一一来看。

**1.默认初始化**

默认初始化是指定义变量时没有指定初值时进行的初始化操作。例如int a; Sales\_data myData;等等。这些变量被定义了而不是仅仅被声明（因为没有extern关键字修饰），而且没有显式的赋予初值。特别的，如果采用动态分配内存的方式（即采用new关键字）创建的变量，不加括号时（如int \*p=new int;）也是默认初始化，加了括号（如int \*p=new int()）为值初始化。变量的值与变量的类型与定义的位置有关系。

（1）对于内置类型变量（如int，double，bool等），如果定义在语句块外（即｛｝外），则变量被默认初始化为0；如果定义在语句块内（即｛｝内），变量将拥有未定义的值。

（2）对于类类型的变量（如string或其他自定义类型），不管定义于何处，都会执行默认构造函数。如果该类没有默认构造函数，则会引发错误。因此，建议为每个类都定义一个默认构造函数（=default）。

**2.值初始化**

值初始化是值使用了初始化器（即使用了圆括号或花括号）但却没有提供初始值的情况。例如，int \*p=new int();vector vec(10);等等都是典型的值初始化方式。注意，当不采用动态分配内存的方式（即不采用new运算符）时，写成int a();是错误的值初始化方式，因为这种方式声明了一个函数而不是进行值初始化。如果一定要进行值初始化，必须结合拷贝初始化使用，即写成int a=int();值初始化和默认初始化一样，对于内置类型初始化为0，对于类类型则调用其默认构造函数，如果没有默认构造函数，则不能进行初始化。

**3.直接初始化与拷贝初始化**

直接初始化与拷贝初始化对应，其内部实现机理不同。直接初始化是指采用小括号的方式进行变量初始化（小括号里一定要有初始值，如果没提供初始值，那就是值初始化了！）。例如int a(12);Sales\_data myData(para);vector ivec(ivec2);string s(“123456”);等等。拷贝初始化是指采用等号（=）进行初始化的方式。例如int a=12;string s=string(“123456”);等等。拷贝初始化看起来像是给变量赋值，实际上是执行了初始化操作，与先定义再赋值本质不同。

（1）对于内置类型变量（如int，double，bool等），直接初始化与拷贝初始化差别可以忽略不计。

（2）对于类类型的变量（如string或其他自定义类型），直接初始化调用类的构造函数（调用参数类型最佳匹配的那个），拷贝初始化调用类的拷贝构造函数。

特别的，当对类类型变量进行初始化时，如果类的构造函数采用了explicit修饰而且需要隐式类型转换时，则只能通过直接初始化而不能通过拷贝初始化进行操作。

**4.列表初始化**

列表初始化是C++ 11 新引进的初始化方式，它采用一对花括号（即｛｝）进行初始化操作。能用直接初始化和拷贝初始化的地方都能用列表初始化，而且列表初始化能对容器进行方便的初始化，因此在新的C++标准中，推荐使用列表初始化的方式进行初始化。列表初始化的应用场景有：int a{12};string s{“123”};vector vec{1,2,3};这里一定要注意，列表初始化使用的是花括号而不是圆括号！

# 重载函数和默认参数的函数的匹配顺序：