

# C++ const对象（常对象）

在 C++ 中，const 也可以用来修饰对象，称为常对象。一旦将对象定义为常对象之后，就只能调用类的 const 成员（包括 const 成员变量和 const 成员函数）了。

定义常对象的语法和定义常量的语法类似：

```
const class object(params);  
class const object(params);
```

当然你也可以定义 const 指针：

```
const class *p = new class(params);  
class const *p = new class(params);
```

class 为类名，object 为对象名，params 为实参列表，p 为指针名。两种方式定义出来的对象都是常对象。

如果你对 const 的用法不理解，请猛击《[C语言const的用法详解](#)》。

一旦将对象定义为常对象之后，不管是哪种形式，该对象就只能访问被 const 修饰的成员了（包括 const 成员变量和 const 成员函数），因为非 const 成员可能会修改对象的数据（编译器也会这样假设），C++ 禁止这样做。

常对象使用举例：

```
01. #include <iostream>  
02. using namespace std;  
03.  
04. class Student{  
05. public:  
06.     Student(char *name, int age, float score);  
07. public:  
08.     void show();  
09.     char *getname() const;  
10.     int getage() const;  
11.     float getscore() const;  
12. private:  
13.     char *m_name;  
14.     int m_age;  
15.     float m_score;  
16. };  
17.
```

```
18. Student::Student(char *name, int age, float score): m_name(name), m_age(age),  
    m_score(score) { }  
19. void Student::show() {  
20.     cout<<m_name<<"的年龄是"<<m_age<<"，成绩是"<<m_score<<endl;  
21. }  
22. char * Student::getname() const {  
23.     return m_name;  
24. }  
25. int Student::getage() const {  
26.     return m_age;  
27. }  
28. float Student::getscore() const {  
29.     return m_score;  
30. }  
31.  
32. int main() {  
33.     const Student stu("小明", 15, 90.6);  
34.     //stu.show(); //error  
35.     cout<<stu.getname()<<"的年龄是"<<stu.getage()<<"，成绩是"<<stu.getscore()<<endl;  
36.  
37.     const Student *pstu = new Student("李磊", 16, 80.5);  
38.     //pstu -> show(); //error  
39.     cout<<pstu->getname()<<"的年龄是"<<pstu->getage()<<"，成绩是"<<pstu->getscore()<<endl;  
40.  
41.     return 0;  
42. }
```

本例中，stu、pstu 分别是常对象以及常对象指针，它们都只能调用 const 成员函数。