

# 1. 重写和重载的区别

## 重载

1. 是指同一可访问区内被声明的几个具有 **不同参数列（参数的类型，个数，顺序不同）** 的同名函数
2. 根据参数列表确定调用哪个函数，**重载不关心函数返回类型。**

```
#include<bits/stdc++.h>
using namespace std;
class A
{
    void fun() {};
    void fun(int i) {};
    void fun(int i, int j) {};
};
```

## 重写（覆写）

是指**派生类中存在重新定义的函数**。其**函数名，参数列表，返回值类型**，所有都必须同基类中被重写的函数**一致**。只有**函数体**不同（花括号内），**派生类调用时会调用派生类的重写函数，不会调用被重写函数**。重写的基类中被重写的函数必须有**virtual**修饰。

```
#include<bits/stdc++.h>
using namespace std;
class A
{
public:
    virtual void fun()
    {
        cout << "A";
    }
};
class B :public A
{
public:
    virtual void fun()
    {
        cout << "B";
    }
};
int main(void)
{
    A* a = new B();
    a->fun();//输出B
}
```

## 重载和重写的区别

1. 范围区别：重写和被重写的函数在不同的类中，重载和被重载的函数在同一类中。
2. 参数区别：重写与被重写的函数参数列表一定相同，重载和被重载的函数参数列表一定不同。
3. virtual的区别：重写的基类函数必须要有virtual修饰，重载函数和被重载函数可以被virtual修饰，也可以没有。

参考来源：[C++中重载和重写的区别\\_菜鸡工坊-CSDN博客\\_c++中的重载和重写的区别](#)

[返回目录](#)