# 重载 覆盖 隐藏

重载：
- 范围相同(作用域相同)
- 函数名相同，参数不同
- 和返回值无关

覆盖：
- 作用域不同(父类和子类)
- 父类有virtual
- 函数名字和参数必须相同

隐藏：
- 作用域不同(父类和子类)
- 基类的函数不是virtual，子类又重新定义了该函数，无论参数相同不相同都是隐藏
- 如果是virtual 函数， 参数相同就是覆盖，参数不同就是隐藏。[参数必须不同]

重载 覆盖 隐藏是C++中最为常见的几种函数相关的概念，特别是在存在虚函数的类继承中特别容易混淆。因此，区分三个概念是非常重要的。

## 重载（overload）

- 重载是在一个类中，相同的函数名，不同的参数，可以实现重载。跟返回值无关，返回值不同，不能叫做重载。
- 不是两个函数的名字相同就能构成重载。
- 全局函数和类的成员函数同名也不算重载，因为函数的作用域不同。例如

```
void Print();//全局函数
class A{
  public:
  void Print();
};
```

- 不论Print的函数是否不同，都不算重载。如果某个类的成员要调用全局函数时候，必须用::Print();

- 还有就是C++是C的一个扩展，如果C++要调用C语言编译后的C函数改怎么办？比如void foo(int x,int y);C语言编译的时候产生的库中的名字是*foo,而C++的编译器会产生*foo_int_int这样的名字来应付重载。由于编译后的名字不同，C++不能直接调用C函数。C++提供了一个C连接交换指定符号extern "C"{ void foo(int x, int y );//或者#include ".h"}来解决

## 覆盖（override）

覆盖发生在虚函数的继承时候发生的，而且发生在不同的范围内，基类和子类。函数前必须有virtual。而重载有没有virtual无所谓。而且**函数名字和参数必须相同**。如果参数不同，就是隐藏，不是重载哦。

## 隐藏（hide）

隐藏发生在继承时候。如果基类的函数不是virtual，子类又重新定义了该函数，无论参数相同不相同都是隐藏。如果是virtual 函数， 参数相同就是覆盖，参数不同就是隐藏。

举个例子：

```
class Base{
public:
    virtual void f(float x);
    void g(float x);
    void h(float x);
};
class Derived{
    void f(float x);//覆盖
    void f(int x);//隐藏
    void g(float x);//隐藏
    void h(int x);//隐藏
};
```