

类外实现类模板函数

//一：普通类的成员函数模板  
//不管是普通类，还是类模板，它的成员函数可以是一个函数模板，称为“成员函数模板”，不可以是虚函数，否则编译器会报错。

```
//A a;  
//a.myft(3); //编译器在遇到这条语句时，编译器就和ui实例化这个函数模板。
```

//二：类模板的成员函数模板

//类模板的模板参数必须用<>指定，成员函数模板(函数模板)的参数都可以推断

```
A<float> a(1, 2);
```

```
A<float> a2(1.1, 2.2);
```

//类模板的成员函数(包括普通成员函数/成员函数模板) 只有为程序所用(代码中出现了对该函数或者该函数模板的调用时)才进行实例化  
//如果某函数从未使用，则不会实例化该成员函数。

```
a.myft(0);  
a.myft(12);
```

I

```
template <typename C> //类的模板参数  
class A  
{  
public:  
    template <typename T2>  
    A(T2 v1, T2 v2): //构造函数我们也引入自己的模板，和整个类的模板C没有关系  
    {  
        template <typename C> //先跟类的模板参数列表  
        template <typename T2> //再跟构造函数自己的模板参数列表  
        A<C>::A(T2 v1, T2 v2)  
        {  
            cout << v1 << v2 << endl;  
        }  
    }  
};
```