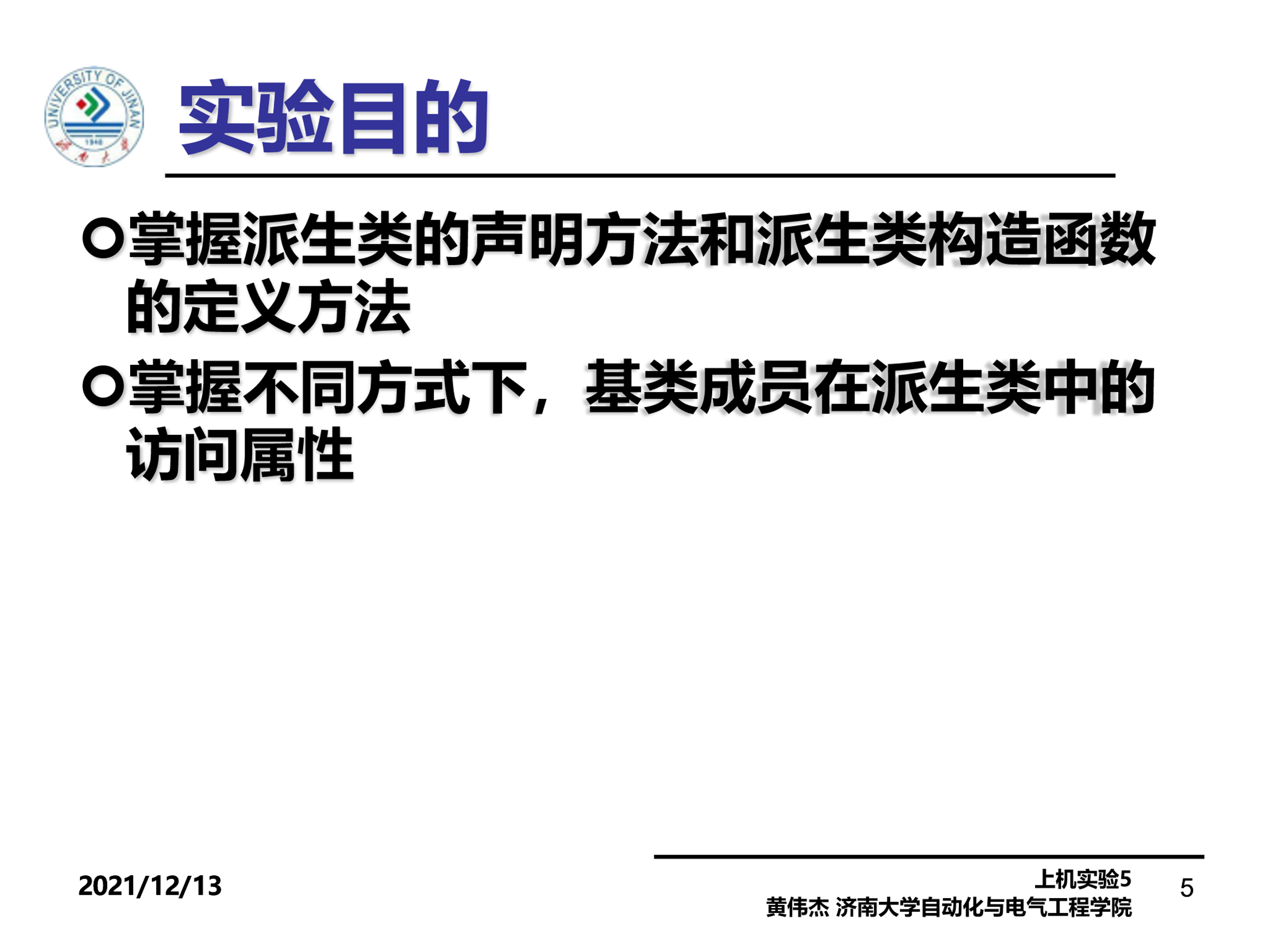
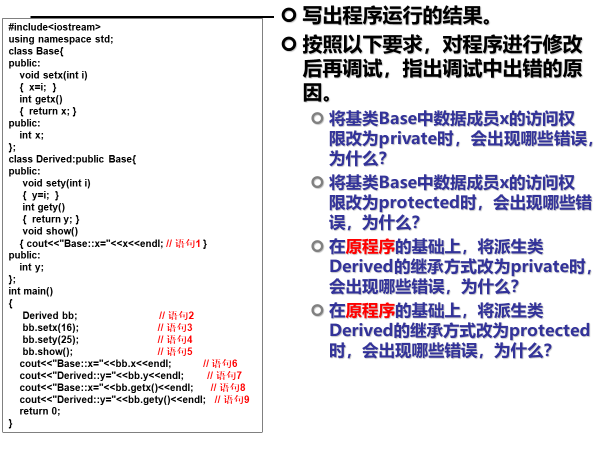
Copyright ©2021-2099 SuDequan. All rights reserved

**实验要求：**



****

**程序代码：**

#include <iostream>

using namespace std;

class Base {

public:

void setx(int i)

{

x = i;

}

int getx()

{

return x;

}

private:

int x;

};

class Derived :public Base {

public:

void sety(int i)

{

y = i;

}

int gety()

{

return y;

}

void show()

{

cout << "Base::x=" << x << endl;

}

public:

int y;

};

int main()

{

Derived bb;

bb.setx(16);

bb.sety(25);

bb.show();

cout << "Base::x=" << bb.x << endl;

cout << "Derived::y=" << bb.y << endl;

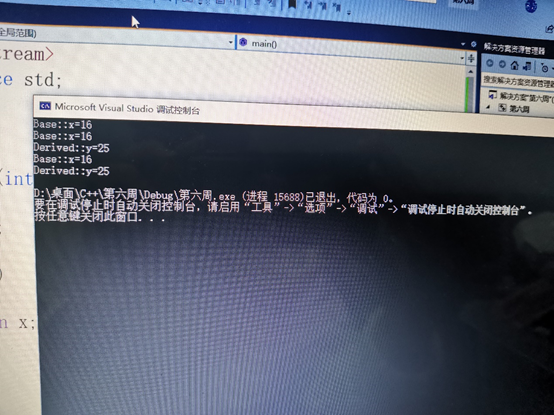
cout << "Base::x=" << bb.getx() << endl;

cout << "Derived::y=" << bb.gety() << endl;

return 0;

}

**程序运行结果如下：**

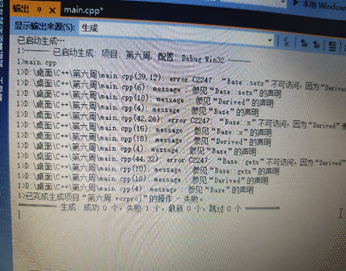


1. 出现错误为：bb.x无法访问，因为类的private成员无法在类外访问。

2. 出现错误为：bb.x无法访问，因为类的保护成员无法在类外访问。

3. 出现错误为：bb.x , bb.getx()无法访问，因为Derived的继承方式为private,类的private成员无法在类外访问。

4. 出现错误为：bb.x , bb.getx()无法访问，因为Derived的继承方式为protected,类的保护成员无法在类外访问。



**感想心得：**

开始时并没有明白是哪里有错误，查阅课本派生类处知识后明白了原因。

派生类构造函数和析构函数的使用原则：

1基类的构造函数和析构函数不能被继承

2若基类没有定义构造函数或有无参的构造函数，派生类也可以不用定义构造函数

3如果基类无无参的构造函数，派生类必须定义构造函数

4如果派生类的基类也是派生类，则每个派生类只负责直接基类的构造

5派生类是否定义析构函数与所属的基类无关

在我看来，基类是比较好的用的一种方式。通过基类的使用，可以找到很多类共用的类。比如说再写列车时，站点和列车都需要时间，时间就可以作为基类，这样做既节省了空间，又让代码更清晰。