程序代码

#include<iostream>

using namespace std;

class Base

{

public:

void setx(int i)

{

x = i;

}

int getx()

{

return x;

}

public:

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;

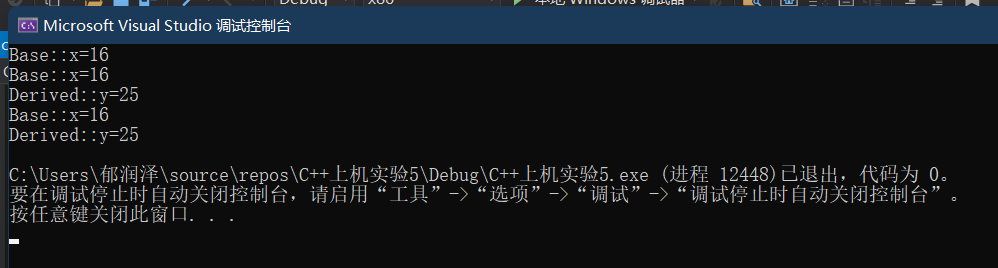
cout << "Base::x=" << bb.getx() << endl;

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

return 0;

}

程序结果

按照以下要求，对程序进行修改后再调试，指出调试中出错的原因。

将基类Base中数据成员x的访问权限改为private时，会出现哪些错误，为什么？文本

描述已自动生成

将数据成员 x 的访问权限改为private后，由于在共有派生类中的成员无法访问基类私有成员

将基类Base中数据成员x的访问权限改为protected时，会出现哪些错误，为什么？图形用户界面, 文本, 应用程序

描述已自动生成

将数据成员 x 的访问权限改为protected后，由于在共有派生类中的成员无法访问基类保护成员

在原程序的基础上，将派生类Derived的继承方式改为private时，会出现哪些错误，为什么？

文本

中度可信度描述已自动生成

将派生类的继承方式改为private时，无论基类是public，private还是protected对象都无法访问

在原程序的基础上，将派生类Derived的继承方式改为protected时，会出现哪些错误，为什么？电脑屏幕截图

描述已自动生成

将派生类的继承方式改为protected时，无论基类是public，private还是protected对象都无法访问

# 程序感想

提高了派生类与继承的的理解能力

Copyright ©2021-2099 RunzeYu. All rights reserved