# 第8周 实验课任务

**1、看程序写结果**

#include <iostream.h>

class Base

{

int i;

public:

Base(int n){cout <<"Constucting base class" << endl;i=n;}

~Base(){cout <<"Destructing base class" << endl;}

void showi(){cout << i<< ",";}

int Geti(){return i;}

};

class Derived:public Base

{

   int j;

Base aa;

public:

Derived(int n,int m,int p):Base(m),aa(p){

cout << "Constructing derived class" <<endl;

j=n;

}

~Derived()

{cout <<"Destructing derived class"<<endl;}

void show()

{Base::showi();

cout << j<<"," << aa.Geti() << endl;}

};

void main()

{ Derived obj(8,13,24);

 obj.show();

}

[参考答案] 说明：派生类的构造函数的执行次序，先调用基类的构造函数，再调用派生类中子对象类的构造函数，最后调用派生类的构造函数。析构函数的执行次序与构造函数正好相反，先调用派生类的析构函数，再调用派生类中子对象类的析构函数，最后调用基类的析构函数。

运行结果：

Constucting base class

Constucting base class

Constructing derived class

13,8,24

Destructing derived class

Destructing base class

Destructing base class

**2、编写一个继承person类的student类。**

#include <iostream>

using namespace std;

#include<string.h>

class person

{

char name[10];

int age;

public:

void init1(char \*str, int k)

{ strcpy(name,str); age=k; }

virtual void disp()

{ cout<<"name: "<<name<<endl;

cout<<"age: "<<age<<endl;

}

};

class student: public person

{ int num;

double avg;

public:

void init2(int i,double f)

{num=i; avg=f;}

void disp()

{ cout<<"num: "<<num<<" avg: "<<avg<<endl; }

};

int main()

{ person A,\*p;

student B;

A.init1("王明",19);

p=&A;

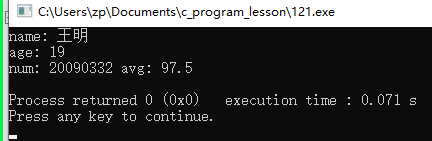
p->disp();

B.init2(20090332,97.5);

p=&B;

p->disp();

return 0; }



**3、声明一个Shape基类，在此基础上派生出Rectangle和Circle类，二者都有GetArea()函数计算对象的面积。使用Rectangle类创建一个派生类Square。**

#include<iostream>

using namespace std;

#define PI 3.1415926

class shape

{

protected:

double s;

public:

void show()

{cout<<"s= "<<s<<endl; }

};

class Rectangle:public shape

{

double x,y;

public:

Rectangle(double x1=0,double y1=0)

{x=x1;y=y1;}

double GetArea()

{s=x\*y;return s;}

};

class Circle:public shape

{

double r;

public:

Circle(double r1=0)

{r=r1;}

double GetArea()

{s=r\*r\*PI;return s; }

};

class Square:public Rectangle

{

double a;

public:

Square(double a1=0 )

{a=a1;}

double GetArea()

{s=a\*a; return s; }

};

int main()

{

Rectangle r(2,3);

cout<<r.GetArea()<<endl;

Circle c(4);

cout<<c.GetArea()<<endl;

Square s(6);

cout<<s.GetArea()<<endl;

return 0;

}

**4、定义一个基类有姓名、性别、年龄，再由基类派生出教师类和学生类，教师类增加工号、职称和工资，学生类增加学号、班级、专业和入学成绩。**

#include<iostream>

#include <iomanip>

#include <string>

using namespace std;

class base //定义一个基类

{

protected:

string name; //姓名

string sex; //性别

int age; //年龄

};

class teacher:public base //基类派生出教师类

{

private:

int sno; //工号

string zc; //职称

double wages; //工资

public:

teacher(int s,string a,double w):sno(s),zc(a),wages(w)

{}

void show()

{cout<<sno<<" "<<zc<<" "<<wages<<endl;}

};

class student :public base //基类派生出学生类

{

private:

int sno; //学号

string bj; //班级

string zy; //专业

double score; //入学成绩

public:

student(int s,string b,string z,double s1):sno(s),bj(b),zy(z),score(s1)

{}

void show()

{cout<<sno<<" "<<bj<<" "<<" "<<zy<<" "<<score<<endl;}

};

int main()

{

teacher t(2345,"professor",5899.0);

t.show();

student s(789,"2021级","软件工程",567);

s.show();

return 0;

}

#include<iostream>

using namespace std;

#include<string>

class Person

{

friend class Teacher;

friend class Student;

private:

string name,gender;

int age;

};

class Teacher:public Person

{

private:

string ID,possetion;

double salery;

public:

Teacher(string n,string g,int a,string i,string p,double s)

{

name=n;gender=g;age=a;ID=i;possetion=p;salery=s;

}

void showinformation()

{

cout<<"姓名："<<name<<"\t"<<"性别："<<gender<<"\t"

<<"年龄："<<age<<"\t"<<"工号："<<ID<<"\t"<<"职称："

<<possetion<<"\t"<<"工资："<<salery<<"\t"<<endl;

}

};

class Student:public Person

{

private:

string num,room,aspect;

double score;

public:

Student(string n,string g,int a,string nu,string r,string as,double s)

{

name=n;gender=g;age=a;num=nu;room=r;aspect=as;score=s;

}

void showinformation()

{

cout<<"姓名："<<name<<"\t"<<"性别："<<gender<<"\t"

<<"年龄："<<age<<"\t"<<"学号："<<num<<"\t"<<"班级："

<<room<<"\t"<<"专业："<<aspect<<"\t"<<"入学成绩："

<<score<<"\t"<<endl;

}

};

int main()

{

Teacher t("张雨佳","女",18,"9420","化学教授",100000);

t.showinformation();

Student s("王路明","男",19,"0249","一年一班","土木工程",10);

s.showinformation();

}

**5、给出下面的基类：**

class area\_cl

{

protected:

double height;

double width;

public:

area\_cl(double r,double s)

{

height=r;

width=s;

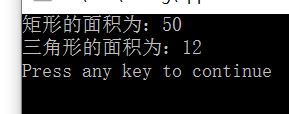
}

virtual double area()=0;

};

要求：

1. 建立基类area\_cl的两个派生类rectangle与isosceles，让每个派生类都包含一个函数area()，分别用来返回矩形与三角形的面积。用构造函数对height和width进行初始化。
2. 写出主程序，用来求height于width分别为10.0与5.0的矩形面积，以及求height与width分别为4.0和6.0的三角形面积。
3. 要求通过使用基类指针访问虚函数的方法（即运行时的多态性）分别求出矩形和三角形面积。



#include<iostream>

using namespace std;

class area\_cl

{

protected:

double height;

double width;

public:

area\_cl(double r,double s)

{

height=r;

width=s;

}

virtual double area()=0;

};

class rectangle:public area\_cl

{

public:

rectangle(double r,double s):area\_cl(r,s)

{}

virtual double area()

{

return height\*width;

}

};

class isosceles:public area\_cl

{

public:

isosceles(double r,double s):area\_cl(r,s)

{}

virtual double area()

{

return (height\*width)/2;

}

};

int main()

{

area\_cl \*p;

rectangle r(10.0,5.0);

p=&r;

cout<<"矩形的面积为："<<p->area()<<endl;;

isosceles i(4.0,6.0);

p=&i;

cout<<"三角形的面积为："<<p->area()<<endl;;

return 0;

}

6、写出程序结果，考察虚函数+对象指针地址

#include<iostream>

using namespace std;

class GA{

public:

virtual int f(){return 1;}

};

class GB: public GA{

public:

virtual int f(){return 2;}

};

void show(GA g){cout<<g.f();}

void display(GA &g){cout<<g.f();}

int main(){

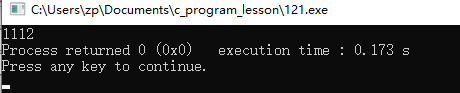
GA a; show(a); display(a);

GB b; show(b); display(b);

return 0;

}

结果：



#include<iostream>

using namespace std;

class B{

public:

virtual void show(){cout<<"B";}

};

class D:public B{

public:

void show() {cout<<"D";}

};

void fun1(B \*ptr){ptr->show();}

void fun2(B &ref){ref.show();}

void fun3(B b){b.show();}

int main(){

B b,\*p=new D;

D d;

fun1(p);

fun2(b);

fun3(d);

return 0;

}

